Software Design Document

Kareem Emad, Nouran Khaled, Shehab Mohsen, Sherif Akram

March 5, 2020

1 Introduction

1.1 Purpose

This software design document purpose is to fully describe the architecture of our Visually Impaired In-Door Assistant: system. Our system depends on people with visual impairments using their smartphones as their eyes to navigate their surroundings. This document will explain in details, the components of the system
represented in the block diagram, the ow of the project with sequence diagram, the data handling in the ER diagram also the implementation of the project and its development will be shown in the class diagram. This software design document (SDD) is, therefore, intended for the stakeholders and developers of our system. This document is also presented as a part of a graduation project at Misr International University (MIU).

1.2 Scope

The system discussed in this document targets end users like people with partial or total impairment that would use Guide Me. Users would get audible directions to their destination as well as warnings if there is too near object that they should avoid in addition to that users shall have their own dataset to save their objects which our system will use to find a targeted object if it's asked for. It will also be beneficial and helpful for researchers and developers that may work on the visually impaired assistance application.

1.3 Overview

The proposed system uses mobile camera to act as the eyes of the blind person, it sends, the system identifies the user using facial identification and retrieves his personalized items from the database if he has any, then a camera stream is opened, the captured stream to the main model which is object detection using a TenorFlow model with a pre-trained dataset of house items, the user then chooses between the system's two main functionalities using speech by translating it through STT[1] either to safely navigate the room or to look for an item he seeks. If the user chooses safe navigation the objects in the frame are detected and distance to reach them is calculated and the user is notified by

speech output using a TTS[2] tool if the object is too close to the other and is blocking their path and where he could move to avoid that obstacle. If the user chooses finding objects, he then is prompted to say the objects name and moves his phone to capture a stream with as many frames as possible and if the object is detected in the frame the mobile vibrates meaning that the object is in that direction, and the closer he gets to the object the more intense the vibration becomes, if the object is not found after a certain time period of searching frames the user is notified by speech that the object is not found. The user can also add his personal items using a video stream of the item and his audio input as a label for it, either through voice assistance or with the help of a human assistant.

1.4 Definitions and Acronyms

Term	Definition
TTS	Text to speech.
STT	Speech to Text
TensorFlow	open source deep learning framework for on-device and off-device inference.

2 System Overview

In order to provide an accurate assistant for a visually impaired person in a well-lit indoor environment by utilizing a smartphone, we developed a mobile application that uses machine learning and image processing, that will be used for the purpose of identifying the user, collecting data and detecting objects in the user's surroundings and categorizing them into generic household items or obstacles. A software will be developed using python, openCV and Deep Neural Networks to work with the data collected from the smartphone's camera, this software will identify the user using facial recognition, also allow the user to search by speech for his desired item and the software searches for it in the streaming frames and measures the distance to them to provide directions for the user to reach the object, it can also be used for warning the user of incoming obstacles and hazards, the user will also add his own personalized items using the camera with the help of a human assistant or voice instructions.

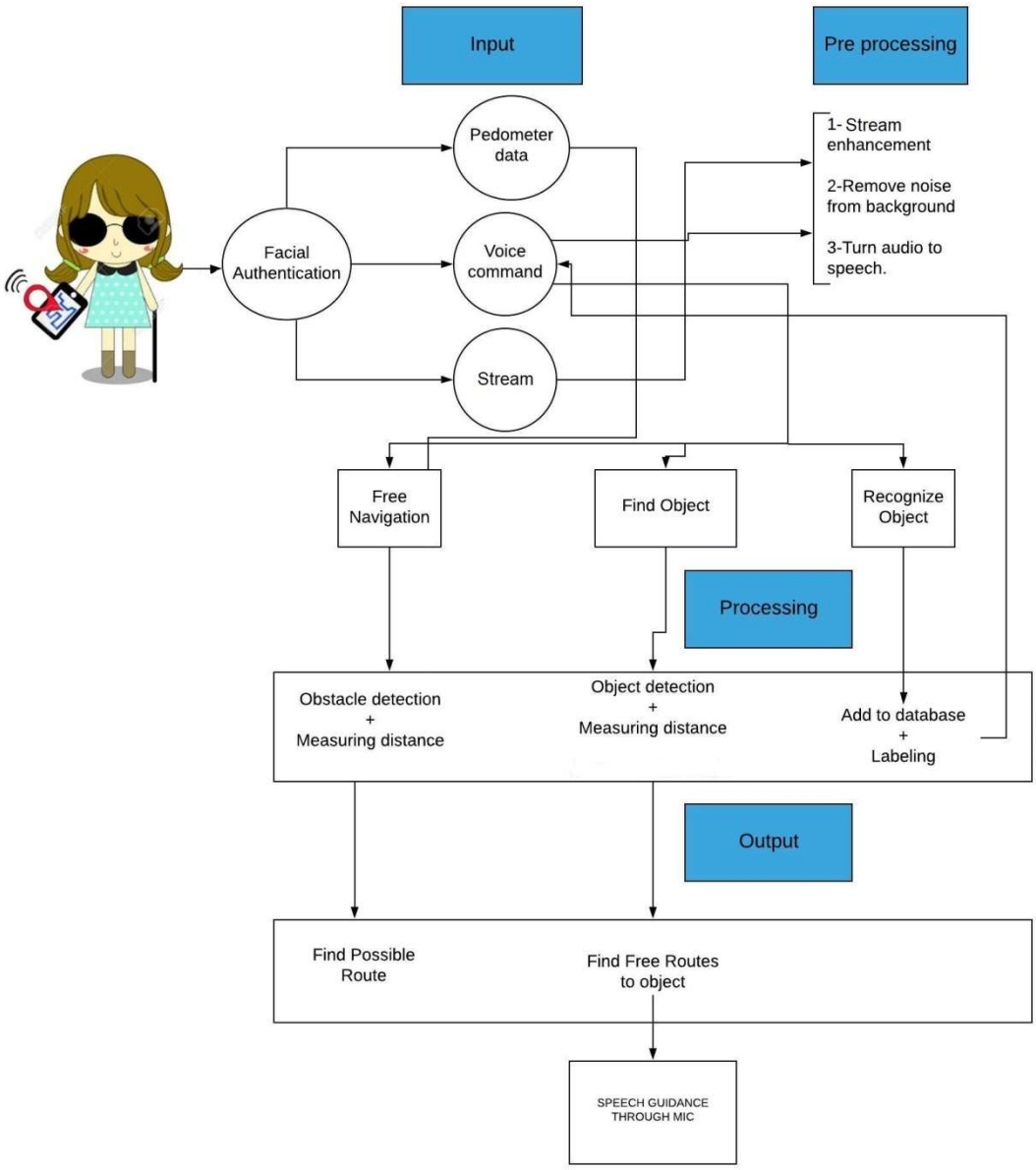

Figure 1: System Overview

3 System Architecture

3.1 Architectural Design

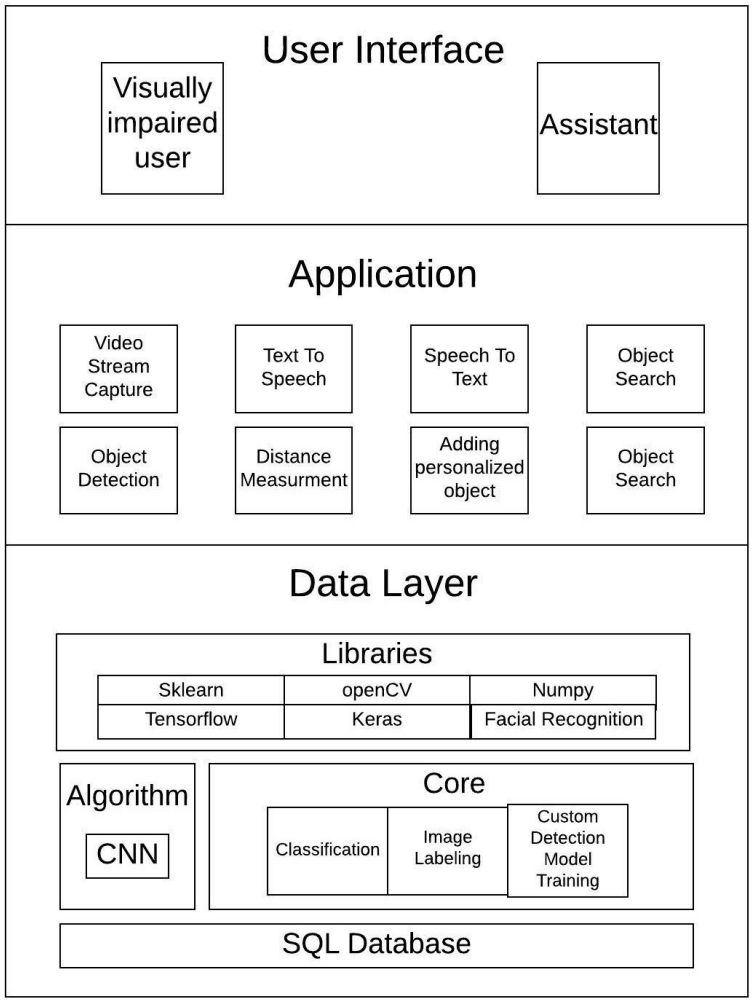

Figure 1: Architectural Design

3.2 Decomposition Description

3.2.1 Class Diagram

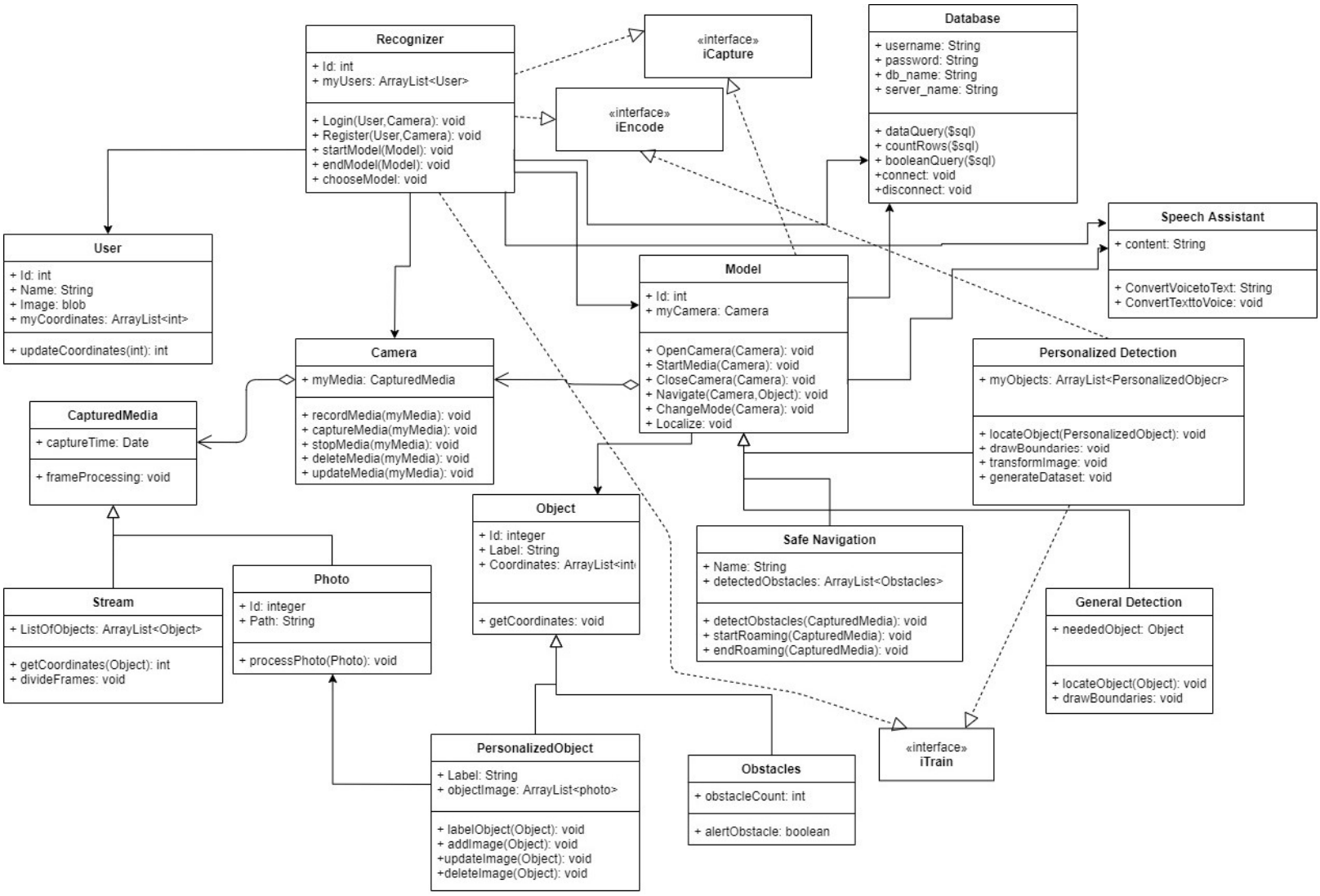

Figure 2: Class Diagram

Sdd/3.2..1.1

Class name: Recognizer

Super class: None

Sub class: None

Purpose: This class controls users and models; it chooses the specified model for detection and connects it with its user

Collaborations: This class aggregates User class and is assisted by Model,Camera,Database and Speech assistant classes

Attributes: id, array list of user

Operations: login (user, camera), void register(user, camera), void start model(model), void end model (model) and void choose model

---

Sdd/3.2..1.2

Class name: user

Super class: None

Sub class:None

Purpose: This class is responsible for handling user information and contains user's face credentials

collaborations: This class is aggregated by Recognizer class

Attributes: id, name, image, array list of my coordinates

Operations: update coordinates (int)

Sdd/3.2..1.3

Class name: camera

Super class:None

Sub class: None

Purpose: This class allows user to use mobile camera and capture media for processing

Collaborations: This class is aggregated by Model class and aggregates CapturedMedia class and it assists Recognizer clas

Attributes: my media

Operations: record media (my media)

Capture media (my media)

Stop media (my media)

Delete media (my media)

Update media (my media)

Sdd/3.2..1.4

Class name: model

Super class: None

Sub class: SafeNavigation,GeneralDetection,PersonalizedDetection

Purpose: The purpose of this class is to allow user to detect object and navigate to object

collaborations: This class assists Recognizer class and it aggregates Camera and Object classes

Attributes: id , camera

Operations: open camera (camera)

Start media (camera)

Close media (camera)

Navigate (camera, object)

Change mode (camera), Localize ()

Sdd/3.2..1.5

Class name: Database

Super class:None

Sub class: None

Purpose: This class purpose is to send queries and execute it in the database and retrieve information

collaborations: This class assists the Recognizer class

Attributes: username, password. Db _name, server_name

Operations:

Data query ($sql)

Count rows ($sql)

Boolean query($sql)

Connect()

Disconnect()

Sdd/3.2..1.6

Class name: speech assistant

Super class: None

Sub class:None

Purpose: This class allows the application to convert user's voice commands into text and convert text to voice instructions

collaborations: This class assists the Recognizer class

Attributes: content

Operations:

convert voice to text ()

convert text to voice

Sdd/3.2..1.7

Class name: personalized Detection

Super class: Model

Sub class: none

Purpose: This class allows the user to detect personalized objects, it trains the detection model and connects the user to his customized model

collaborations: This class inherets the Model class and is assisted by Stream class

Attributes: array list of personalized objects

Operations: locate object personalized object()

Draw boundaries()

Transform image()

Generatedata set ()

Sdd/3.2..1.8

Class name: object

Super class: none

Sub class: obstacles, personalized object

Purpose: This class purpose is to allow model to differentiate between different objects after detection by giving labels and getting object's coordinates

collaborations: This class is aggregated by Model class

Attributes: id, label and array l ist of coordinates

Operations: void get the coordunates

Sdd/3.2..1.9

Class name: obstacles

Super class: object

Sub class: none

Purpose: This class allows the model to differntiate between objects and obstacles after measuring distance to user

collaborations: This class inherets object class

Attributes: obstacle count

Operations: alert obstacle

Sdd/3.2..1.10

Class name: personalized object

Super class: object

Sub class: none

Purpose: This class purpose is to allow user to add photos of new object and give it a label so it can be processed by custom detection model

collaborations: This class inherits object class

Attributes: label, object image

Operations: label object (object)

Add image(object)

Update image(object)

delete image(object)

Sdd/3.2..1.11

Class name: general detection

Super class: Model

Sub class: none

Purpose: This class is responsible for general object detection; it detects objects in stream, draws a boundary box around the object and measures distance to object.

collaborations: This class inherits model class

Attributes: need object

Operations: locate object (object)

Draw boundaries

Sdd/3.2..1.12

Class name: photo

Super class: CapturedMedia

Sub class: none

Purpose: This class allows the user to capture face image for login and registration and allows user to capture object image for custom detection

collaborations: This class is aggregated by PersonalizedObject class and User class

Attributes: id, path

Operations: process photo(photo)

Sdd/3.2..1.13

Class name: stream

Super class: CapturedMedia

Sub class: none

Purpose: The purpose of this class is to capture a stream and send it to the model for object detection processing

collaborations: This class assists the GeneralDetection class

Attributes: array list of object

Operations: get coordinates (object)

Divide frames ()

Sdd/3.2..1.14

Class name: captured media

Super class: none

Sub class: Stream,Photo

Purpose: The purpose of this class is to allow user to capture media in different types to allow detection

Collaborations: This class is aggregated by Camera class

Attributes: capture time

Operations: frame processing ()

3.2.2 Sequence Diagrams

3.2.2.1 Registration Sequence

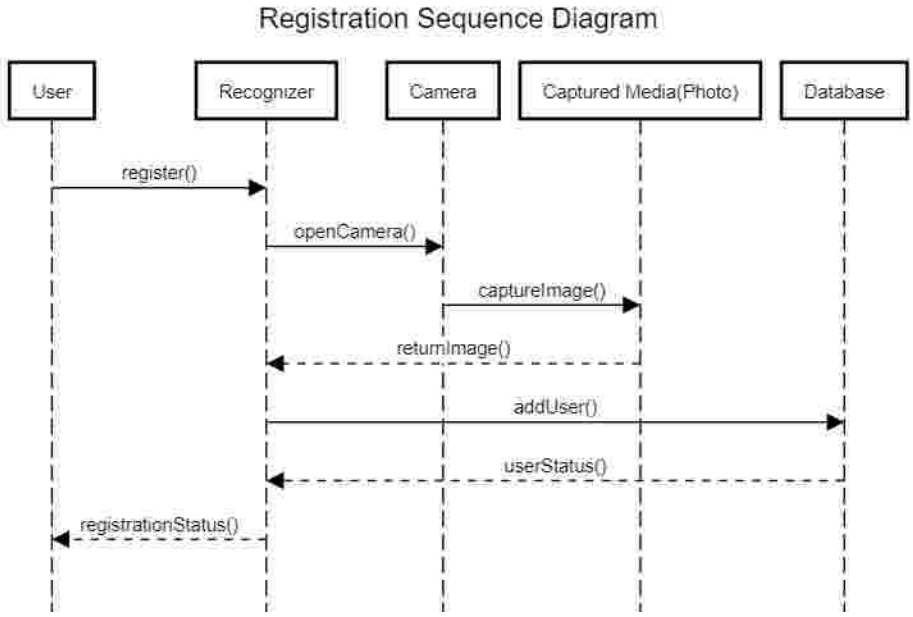

Figure 3: Registration Sequence Diagram

3.2.2.2 Adding Personalized Object Sequence

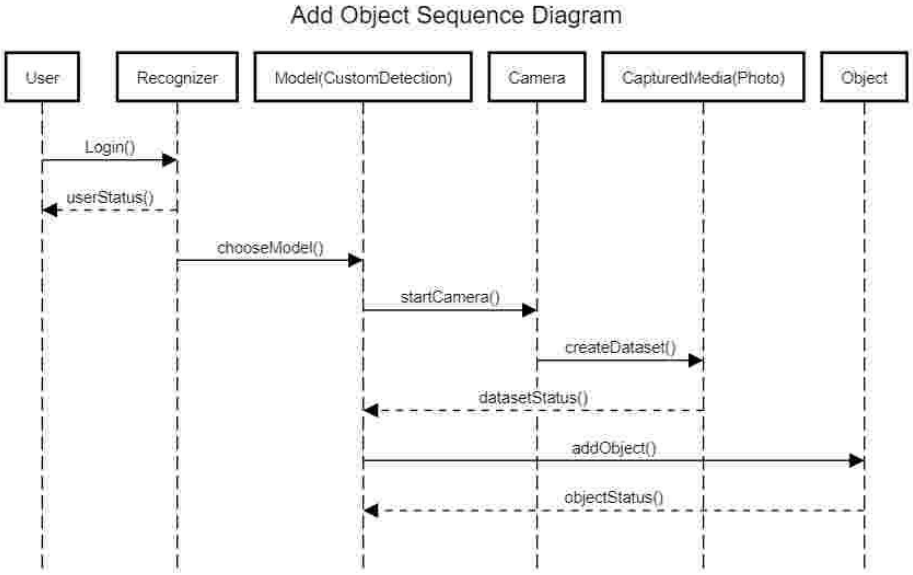

Figure 4: Add object sequence diagram

3.2.2.3 Safe Navigation Sequence

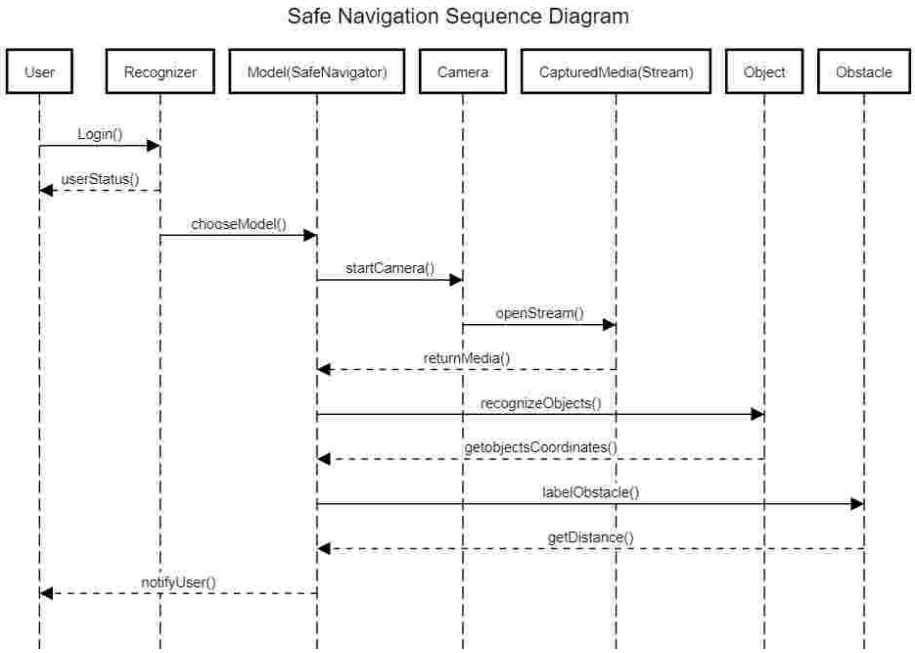

Figure 5: Safe Navigation sequence diagram

3.2.2.4 Train Customized Model Sequence

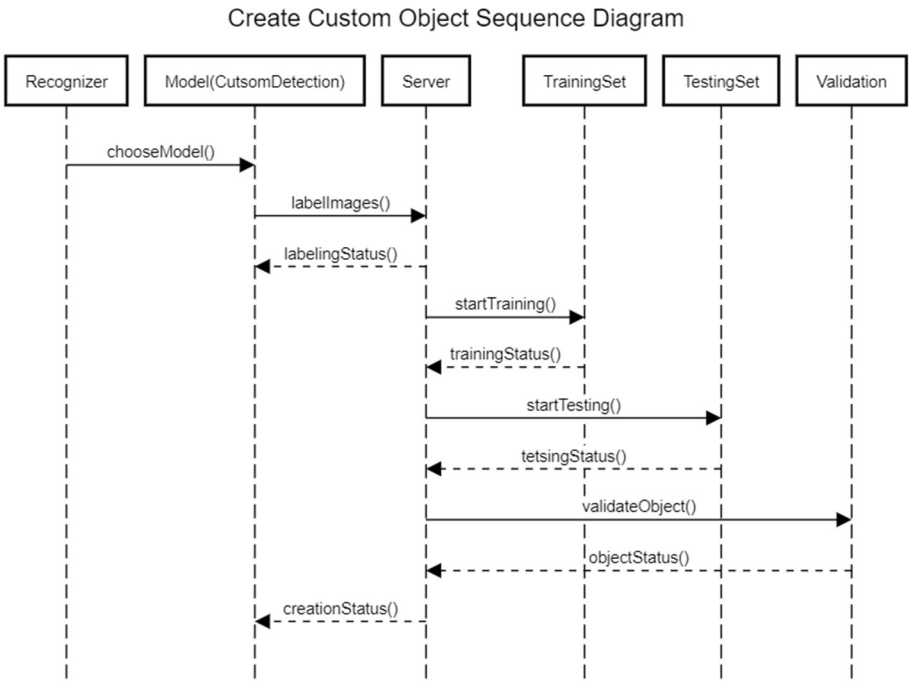

Figure 6: Train customized object sequence diagram

3.2.3 Process diagram

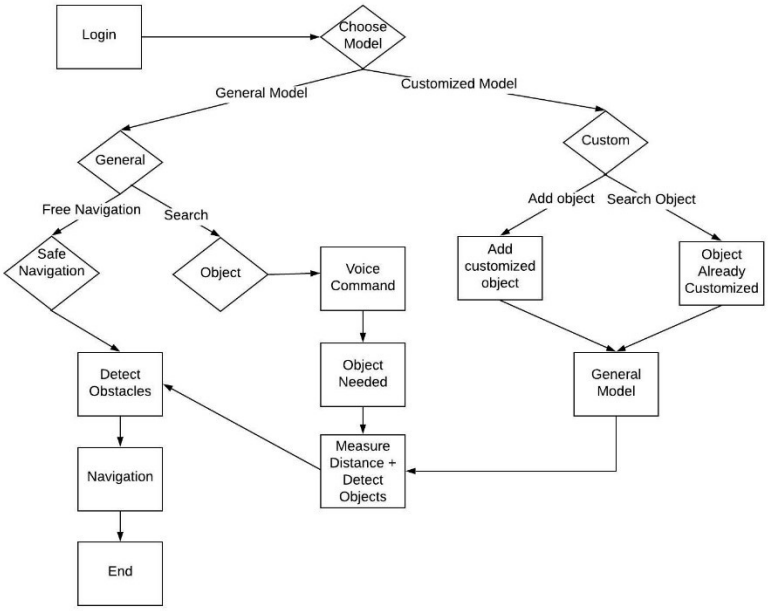

Figure 7: Process diagram

3.3 Design Rationale

3.3.1. N-Layered Architecture

This architecture[3] has been chosen to develop our system as it separates the logic of the layer from the presentation of it which makes each layer independent from the others and as a result a change in a layer won't change in the rest of the layers which makes it easier to implement and to change.

3.3.2. CNN

Convolutional neural network consists of several different layers[4] such as the input layer, at least one hidden layer, and an output layer. They are best used in object detection for recognizing patterns such as edges (vertical/horizontal), shapes, colors, and textures. That's why we chose it in our project to apply along with tensorflow for optimum results in object detection

4 Data Design

4.1 Data Description

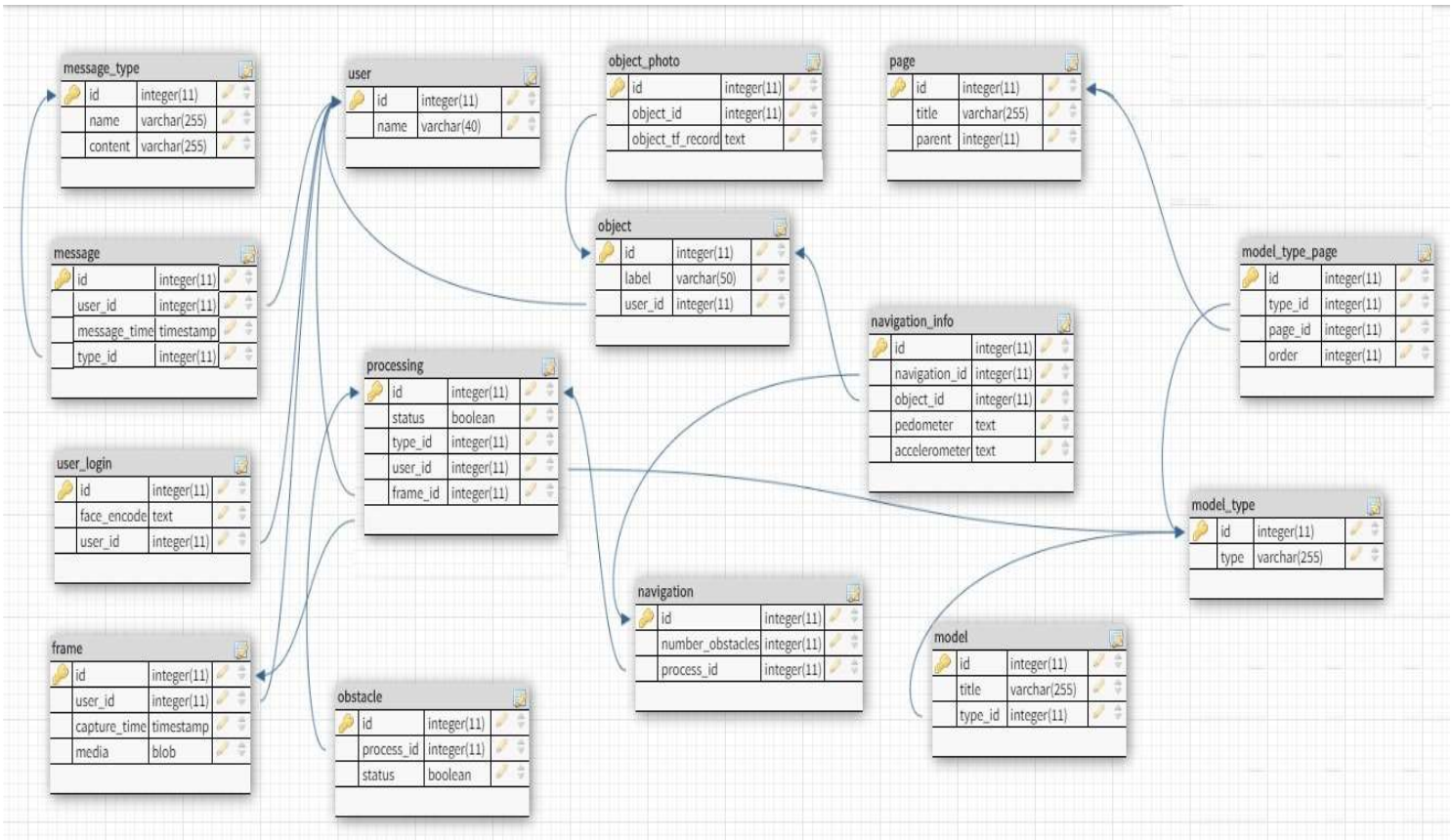

Figure 8: Database diagram

5 Component Design

5.1 Dataset

5.1.1 Luxand dataset:

Luxand FaceSDK [5] returns the coordinates of all human faces that appear in the picture or notifies if no face is found. FaceSDK can track all the faces appearing in a video stream. It also allows finding out if a newface appears in the frame, or if one of the subjects leaves the frame. This in turn enables easy implementation of people counting.

5.1.2 Coco dataset:

Common Objects in Context (Coco)[6] dataset, which contains around 123,287 images, is a large-scale object dataset made by tensorflow to detect common objects. This dataset is used with all of the suggested modules.

5.1.3 Customized dataset:

The following dataset items vary according to each user, since that it's made of the objects that the user wants to recognize the user's assistant takes 30 seconds video of the desired object from different angles, the system then transforms them to (300*300) width and height coordinates, in order to start the feature extraction process. Finally, the dataset record is generated and linked with the user id, which was previously generated from the face authentication module.

5.2 Data Pre-processing

5.2.1 Audio To Speech Conversion

We used Google Speech-to-Text which enables us to convert audio to text by applying powerful neural network models in an easy-to-use API. The API recognizes 120 languages and variants to support global user base.

5.2.2 Customized dataset creation

The user takes a 30 seconds video for the object he/she wants to add the video is then sent to the server and divided into frames to be saved as a collection of images, then these images are resized to (300*300) width and height coordinates, then each frame is labelled

5.3 Processing and Classification

5.3.1 Face Authentication (Model 1):

The user's face is scanned when he signs up for the first time, each face has a unique id which will be saved in database whenever the user adds new object in the customized dataset explained below in model3, these data will be retrieved with id generated from the authentication module.

5.3.2 General Object Detection (Model 2):

The model is trained on the coco dataset using quantized mobilenet sdd[7] the dataset used contains only indoor objects. When an image is subsequently provided to the model, it'll yield a list of the objects it identifies, the area of a bounding box that contains each object, and a score that shows the certainty that discovery was rectified.

5.3.3 Customized Model (Model 3):

After labelling the dataset is then split into 80 percent for training, and the other 20 percent are for testing. Then the TFRecords are created that can be served as input data for training of the object detector, so it is needed to extract features of the object from each image and convert it to a TFRecord. Finally, everything is in place and ready to train the model using quantized mobilenet sdd classifier. To make this model run on a mobile, it is important to start by creating a TensorFlow frozen graph that can be used with TensorFlow lite, then convert the frozen graph to the to the TensorFlow Lite flat buffer format, and finally save it in the database to be ready for use along with user id.

5.3.4 Distance Measuring Navigation (Model 4):

In the process of object detection, the width of each detected object is collected, the distance between the user and the object is then calculated and detected through utilizing the triangle similarity. The formula for measuring the distance is D = (W x F)/P where D is the distance to the object, F is the focal length of the camera, W is the real width of the object and P is the collected width of the object in pixels, The model calculates the distance between the user and the other detected objects, if object pixel width is more than 80% of screen size, the system will notify the user by producing a sound alert to avoid a crash event.

6 Human Interface Design

6.1 Overview of User Interface

The user interactions with the system will be authorized using facial recognition, and his choices will be captured through audio input and the system will present his results using audio output, thus resulting in an easy and convenient way for the visually impaired population to use this application, although the personalization module can be used by a human assistant using buttons if needed.

6.2 Screen Images

6.2.1 General Object Detection

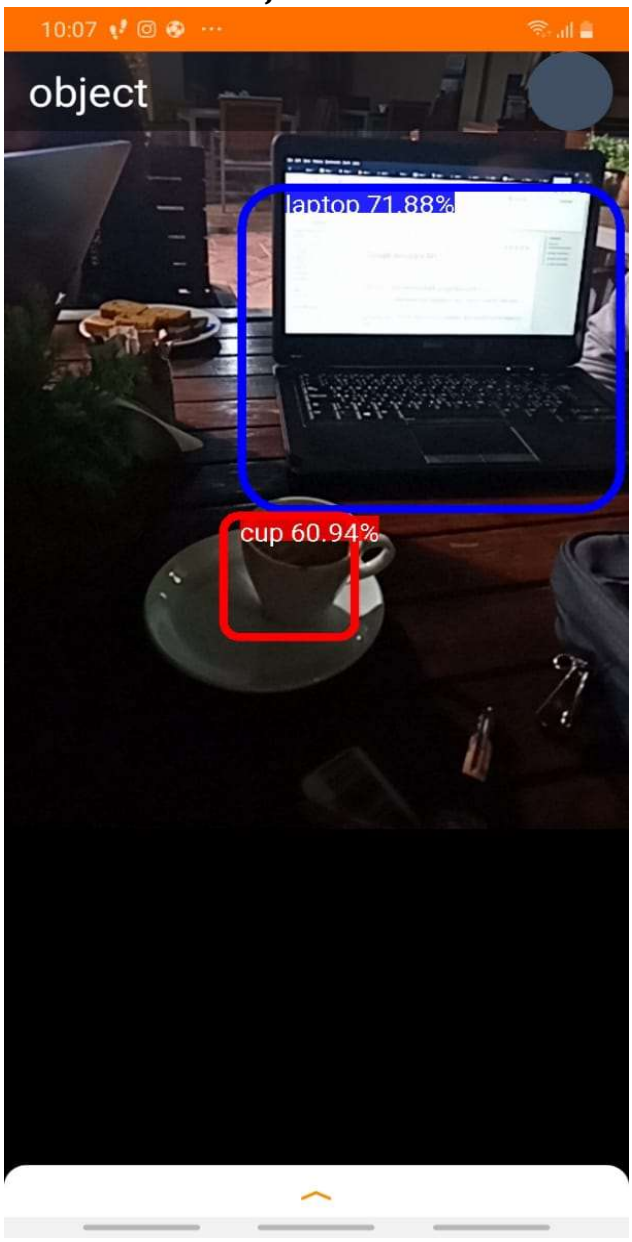

Figure 9: General object detection screen

6.2.2 Customized Object Detection

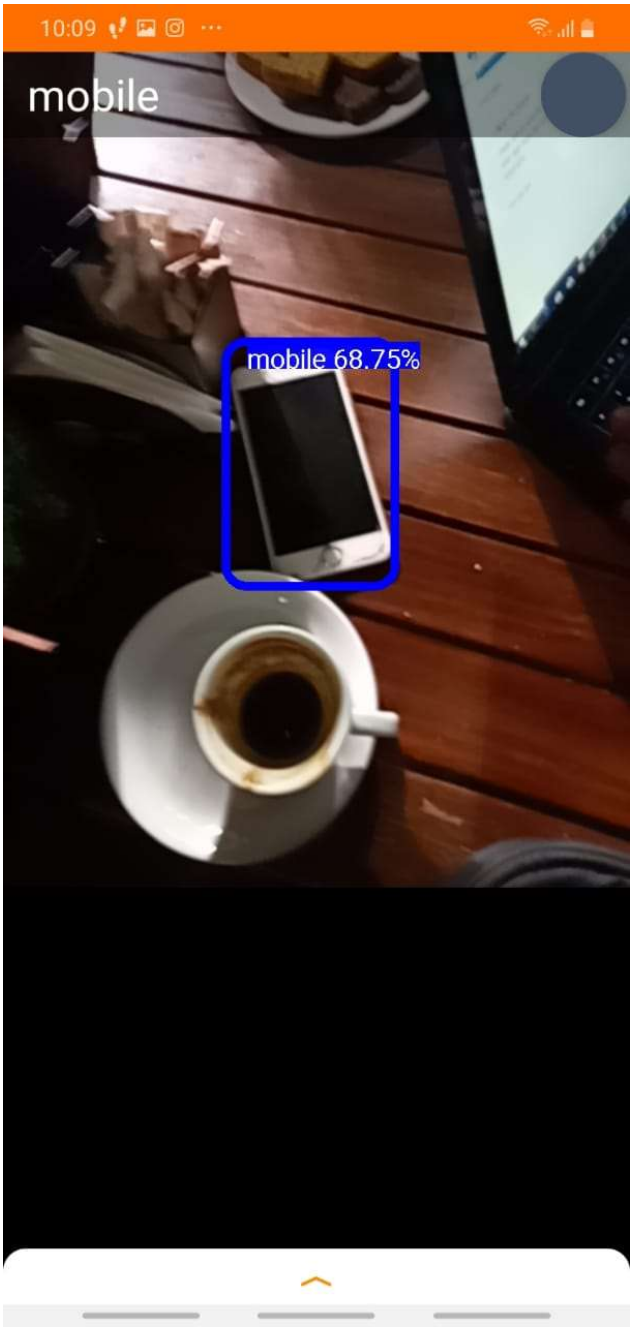

Figure 10: Customized Object Detection

6.2.3 Adding Objects

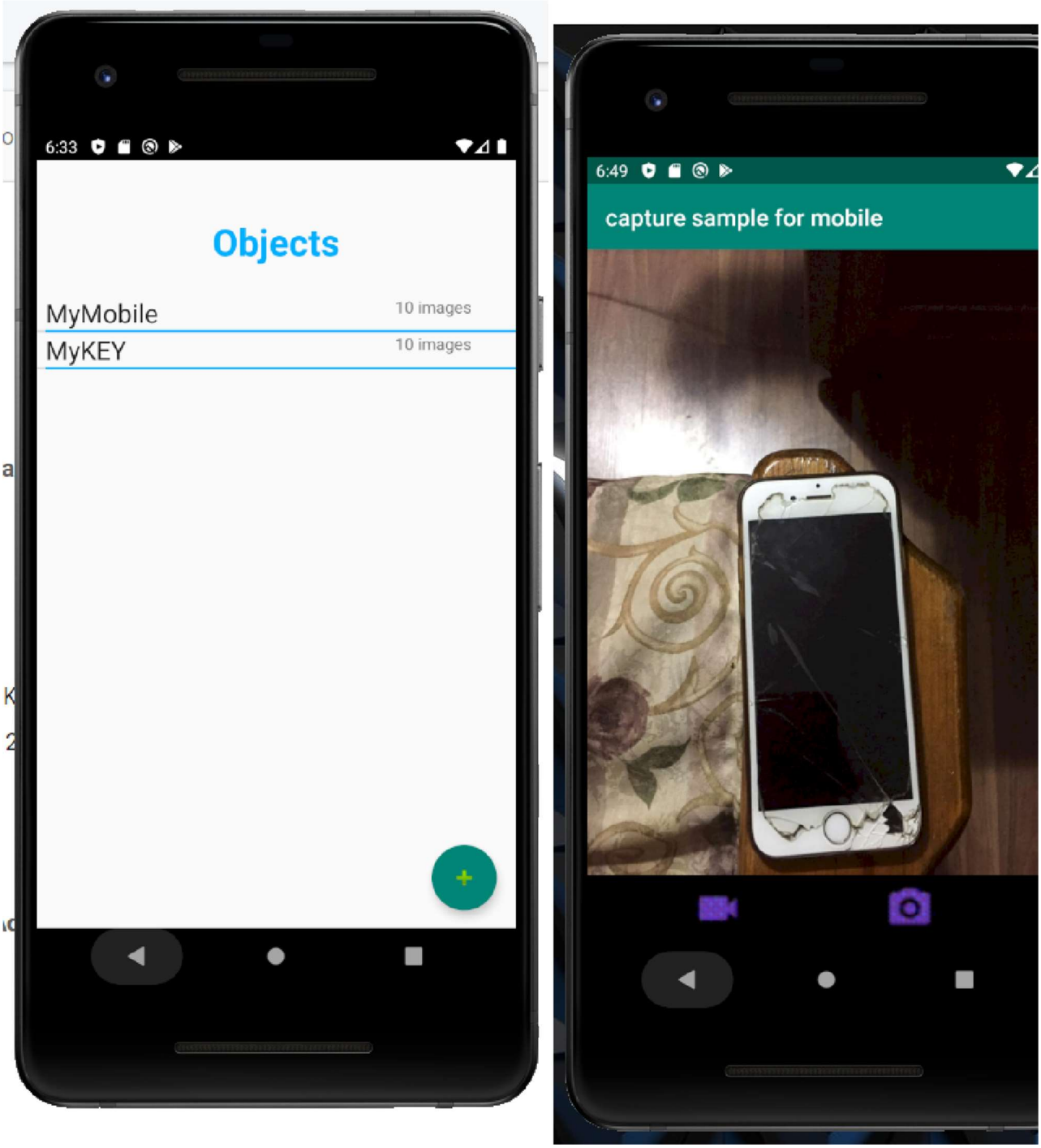

Figure 11: Add object screen

The main interface in our application is a regular camera view, a stream from the front camera is used for facial identification for the user, The system presents the options available at the screen using audio when the user touches any part of the screen, The navigation module uses the back camera to capture a stream of the surroundings, and the Adding objects screen can either be interacted with using audio if the user is the one adding his customized item or through buttons if the user is getting human assistance.

7 Requirements Matrix

Req. ID	Req. Type	Req. Name	Req. Description	Module	Statues	Req. Reference
Fr1	Required	Register	User can register a new account	User	Completed	Sequence diagram
Fr2	Required	Login	The user can login into his account from any device	User	Completed	Sequence diagram
Fr3	Required	Facial features extraction	The system converts facial features to data.	System	Completed	Sequence diagram
Fr4	Required	Live stream	User open a stream for detection	User	Completed	Class Diagram
Fr5	Required	Object detection	Detect and track objects in the stream	System	Completed	Database Diagram
Fr6	Required	Capture images	User captures 10 images for the object he want to add.	User	Completed	Sequence diagram

Req. ID	Req. Type	Req. Name	Req. Description	Module	Statues	Req. Reference
Fr7	Required	Create bounding box	Detect objects in the photo and draw a boundary box.	System	Completed	Class Diagram
Fr8	Required	Generate dataset records	Dataset records (Tensorflow) that can be served as input data for training of the object detector.	System	Completed	Database Diagram
Fr9	Required	Train model	Train user's customized model with customized object.	System	Completed	Sequence diagram
Fr10	Required	User Positioning	The system tracks the user's location compared to the object	System	Completed	Class Diagram
Fr11	Required	Audio Menu	The system presents menu options using speech.	System	Completed	Class Diagram
Fr12	Required	Voice commands	The system converts audio speech to text	System	Completed	Class Diagram
Fr13	Required	Navigation	The system finds and converts the path into audible directions.	System	In Progress	Sequence Diagram

8 References

[1] C. C. N. T. Treephop Saeteng, Traipop Srionuan, "Automatic fingersign-to-speech translation system," Journal on Multimodal User Interfaces, 5 July 2011.

[2] Nwakanma Ifeanyi, Oluigbo Ikenna, Okpala Izunna,"Text – To – Speech Synthesis (TTS)", International Journal of Research in Information Technology (IJRIT), May 2014.

[3] Mark Richards, Software Architecture Patterns. New York:John Wiley and Sons, 2019.

[4] S. J. S. J. Varsha Sharma, Chaitanya Sharma, "Assitance application forvisually impaired - vision,"International Journal of Scientific Researchand Engineering Development, p. 4, November 2019.

[5] A. Samkaria, "Object detection using convolutional neural networks intensor flow,"INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCHIN TECHNOLOGY, vol. 5, pp. 351–353, September 2018

[6] P. S. A. G. Laviniu Ţepelea, Virgil Tiponuţ, "Enhanced Face Verification and Image Quality Assessment Scheme Using Modified Optical Flow Technique"International Journal of Scientific Researchand Engineering Development, p. 4, November 2015.

[7] E. Y. E. K. J. E. H. M. M Tsung-Yi Lin Michael Maire, "Microsoft COCO: Common Objects in Context"2018 9th IEEE An-nual Ubiquitous Computing, Electronics Mobile Communication Conference(UEMCON), p. 6, 2018

[8] R. Y. F. B. Abbas Riazi, Fatemeh Riazi, "Outdoor difficulties experiencedby a group of visually impaired iranian people,"journal of currentophthalmology, p. 85–90., 2016.

[9] . T. C. O. K. K. Manabu Shimakawa, Kosei Matsushita, “Speed/accuracy trade-offs for modern convolutional object detectors”ACIT 2019: Proceedings of the 7th ACIS International Conference onApplied Computing and Information Technology, p. 6, May 2019

[10] E. Y. E. K. J. E. H. M. M. Milios Awad, Tarek Mahmoud, “Intelligent eye:A mobile application for assisting blind people,”2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference(UEMCON), p. 6, June 2018