Software Requirement Specification Document for Visually Impaired Assistant

Mayar Osama, Salma mohamed, Alaa Yehia, Rawan Sherif
Supervised by: Dr. Alaa Hamdy
Assistant supervisor Eng. Noha Elmasry

June 4, 2021

Version	Date	Reason for Change
1.0	21-Dec-2020	SRS First version's specifications are defined.
2.0	26-Dec-2020	Second version includes: Updated class diagram Database schema Edited Functional Requirements

Table 1: Document version history

GitHub: https://github.com/maayarosama/Visually-Impaired-Assistant

Abstract

Our aim is to facilitate the life of visually impaired people who could lack the technology to help them in their lives. This project focuses on the development of a mobile application that uses voice commands and text to voice technology to enable users to interact with the mobile application. The user captures the item they want to identify using the smartphone's camera or can browse through their gallery and pick the image they want to upload for processing. The image chosen then will undergo processing depending on the feature the user chose through the given voice command, the features include food recognition, banknote recognition, text to voice and colors/patterns identification. As well, the application allows the user to register or login using facial recognition to further facilitate the experience for users. The application is designed to be effective and can be easily navigated by visually impaired users.

1 Introduction

1.1 Purpose of this document

The purpose of this document is to explain in detail the requirements needed to develop a mobile application that assists visually impaired people with doing simple daily activities. The application aims to provide an easy and reliable way for visually impaired users to navigate through their daily lives. This document is the basis for every developer who is a part of the development team, will need in order to fulfill all requirements [1]. The document will outline the functional and non-functional requirements as well other factors that collectively covers the development process of this application.

1.2 Scope of this document

This document targets software developers who are part of the developing team on this project. It will contain an overview of our product and a simple explanation of the application so it can give developers a clear idea without the need for future re-design [2]. As well, it will describe the functionality the application shall fulfill.

1.3 System Overview

Provides a brief overview of the product defined as a result of the requirements elicitation process.

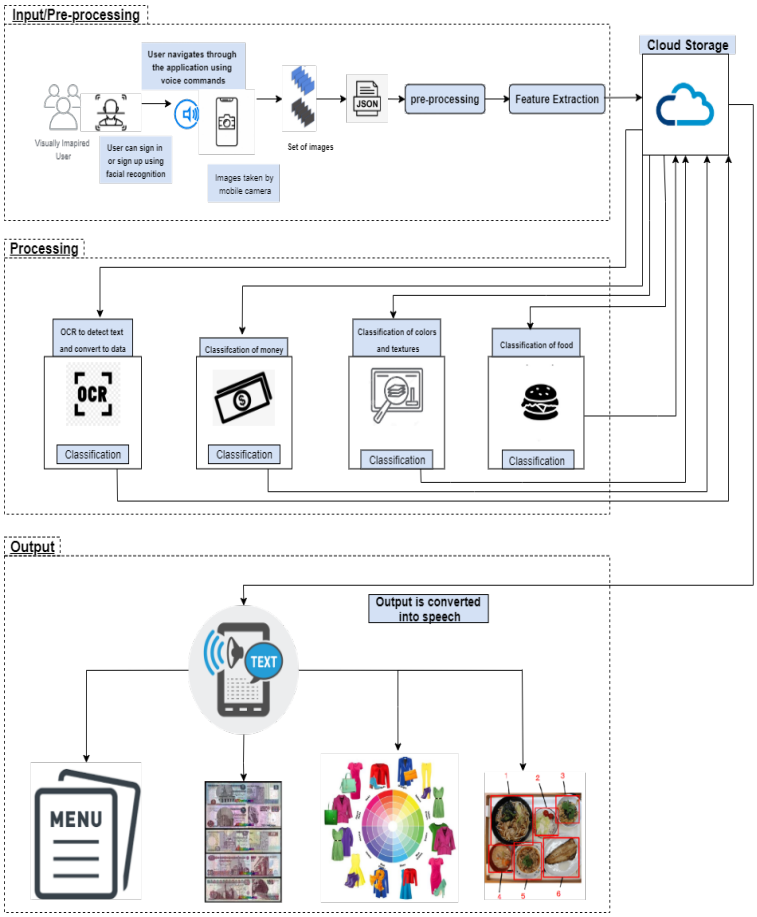

Figure 1: System overview

1.4 System Scope

The system covers the following things:

- Authenticate users using facial recognition
- Allow users to register using facial recognition
- Speech recognition is used to fill in the information spoken by the user at registration
- Users can use specific voice commands to execute the necessary action
- Convert text present in images to audio
- Identify the amount of banknotes present in the image provided by the user
- Recognize food present in the image

- Recognize colors and patterns
- User can edit information present in their profile using speech recognition
- Admin can edit, delete, change and update the data of the users
- Processed data from the images are converted to sound

1.5 Business Context

This project is inspired by the problems many visually impaired people face, Al Nour and Amal's school for visually impaired girls provided insight into the issues many of the younger girls face. The girls present at the school are part of the targeted end users of this application, therefore the issues they face are within the scope of the many issues visually impaired people face in general. The application will help the girls navigate their day to day lives with minimum to no help from other people. As well, the application shall be designed to be easily interacted with to offer maximum accessibility to the users.

2 Similar Systems

2.1 Academic

A Smart Personal AI Assistant for Visually Impaired People[3]: Visually impaired continuously suffer because they have no idea about what are their surroundings. This system was made to assist visually impaired users to know what surrounds them. In this system Machine Learning, Artificial Intelligence, Image, and Text recognition have been used. This system is implemented as an Android mobile application includes voice assistant, image recognition and currency recognition. This application helps the visually impaired user to identify the objects surrounding him. This application took inspiration from another project which has made object detection for the visually impaired by using the camera of a smartphone only[4]. Visually impaired will definitely suffer from taking a good image for the processing of the image for object recognition. So the application took inspiration from another system[5] in which an array of frames will be taken and the best frame will be chosen among them. Also, this application provides the user with a service named chat-bot in which the visually impaired user can ask by voice or text this service anything and it will give him a reply using audio. The algorithms used in this system are decision tree and information gain. The big disadvantage of this application that it's Android only.Besides that this application takes a lot of processing unlike this system[6] in which it takes a lot of frames and chooses the best of them to enter another processing process.

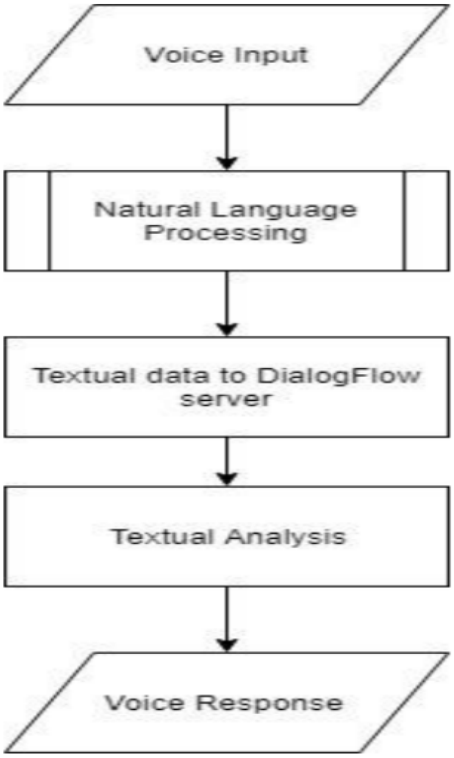

Figure 2: Chat-bot Flowchart

Android-based blind reader system[7]: This application is implemented to help the visually impaired with some problems they face. It implements a Graphical user interface and Auditory user interface The system consists of four levels. The first level contains a Graphical user interface that is mainly used for easier user interaction and another interface which is an Auditory user interface is used for directing users how to use the application right. The second level is the function modules which include three main functions which are Paper reading, electronic reading, and audio notes. The paper reading function has two modes which are default and automatic. In the automatic mode, the system continuously processes images which include extracting text area boundaries by image projection analysis method and checking whether the text is in an effective area and if it's not, the system directs visually impaired users to direct the camera till a qualified image is captured. Electronic reading is to read documents by sound. Audio notes allow the visually impaired user to take notes by audio to check it later on. The third level is image pre-processing, this level to approve if the image is suitable for optical character recognition or not. The fourth level is the optical character recognition and text-to-speech engine, the optical character engine is responsible for recognizing the text in the captured images while text to speech engine is responsible for taking this text and converting it into audio. The disadvantage of this system that it works only with android devices. This system took inspiration from several systems like a non-visual calculator[8] which works by the number of taps and not buttons and also another system which is a braille reader system[9] but it needs an external braille device. The advantage of this application that it doesn't need any external device unlike the braille system[9].

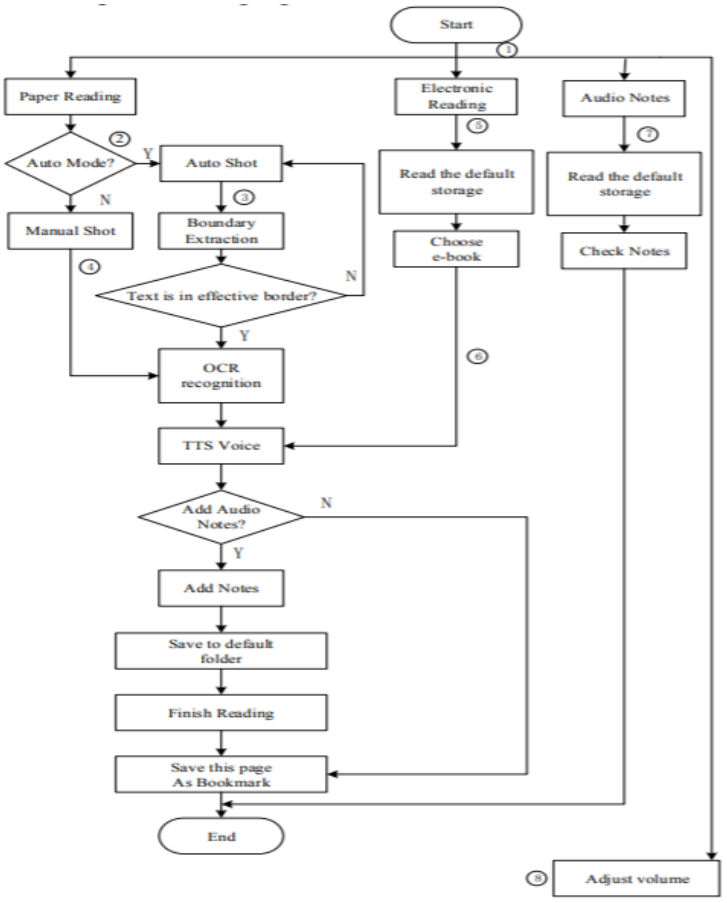

Figure 3: System functions flowchart

The dataset used to test this application is images of pages of 16 Chinese books. From these books, they've only tested certain pages. As the number of operations increases, the number of success rate increases. The biggest disadvantage of this application that it is an android based application so that it doesn't work on a non-android application. On the other side, there is another system named Space anomalies in Arabic [10] that uses the Arabic dataset in OCR to solve the spaces problem to increase the accuracy of Arabic OCR.

Recognizing clothes patterns and colors for blind people[11]: This is a system that helps the visually impaired to identify the colors and patterns of clothes. This system uses a dataset containing 627 images divided into striped, plaid,pattern-less, and irregular. The system can also identify eleven different colors which are red, orange, yellow, green, cyan, blue, purple, pink, black, grey, and white. This system took inspiration from more than one system regarding texture like Texture Analysis using shortest paths in graphs[12] and also Texture representations using subspace embeddings[13]. The image is taken through a camera then the pattern is classified through a support vector machine algorithm. The following diagram explains the flow of the data in the system.

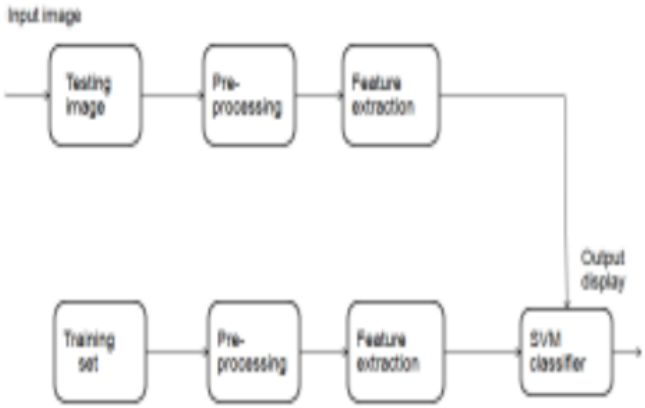

Figure 4: System Block Diagram

The basics The features are extracted using three descriptors like Radon signature descriptor which is used to extract statistical features, and wavelet sub-bands are used to extract global features After the descriptors are done with their jobs, The support vector machine classifies the images according to their categories. Also, this idea was made by another system which is Recognition and Identification of the Clothes in the Photo or Video Using Neural Networks[14]. This system used a convolutional neural network and it was trained on the Clothes image dataset. They expanded their dataset by augmentation methods. The algorithms they used are SIFT and SURF. The final accuracy that this system nailed was about 84%.The disadvantage of this system that it used a small dataset that consisted of 627 images only of different textures and colors so there was not a big variety to test the system on. Better a large dataset should have been used like [15]

Recognition of Thai Banknotes[16]: This is an application that helps visually impaired users recognize Thai banknotes. This system mentioned that it took inspiration from several systems like analyzing the mental model of blind users[17] and also the mental model of blind users to assist designers[18] in which both of them presented a mental model to serve blind user needs. Five people including students and teachers from the school where the the application was examined in was recruited to collect data to see how they deal with smartphones.

Participants	1	2	3	4	5
Sex	M	F	F	M	F
Age	37	31	16	17	16
Position	teacher	teacher	std	std	std
Experience with smartphones	5	3	2	2	1
Experience with smartphone application	SayText, LookTel, Barcode Reader, CamFind	Barcode Reader, CamFind ,Voice Eye	SayText , Barcode Reader	Barcode Reader	Barcode Reader
Blindness type	Totally blind	Totally blind	Totally blind	Totally blind	Totally blind

* M: Male, F: Female, Std: Student

Figure 5: Visually impaired users dealing with smartphones

The prototype was developed by Android studio tools version 1.4 with JDK 1.7. Java was the main programming language used. Also, Text to speech engine was used to convert from any text to an audio message. The device used with this application was LG L70 with 8GB ROM and a 5MP Camera. The dataset used with this system is all Thai Banknotes which is considered very little to be trained on and to be trusted as a system. On the other hand, there is another system that used four different currencies which are the English pound, Japanese Yen, Indian rupee, and Euro. This system which is Paper Currency recognition for color images[19] is made up of four modules which are Image acquisition, pre-processing including noise removal, feature extraction, classification, and recognition. The following diagram shows how this system works.

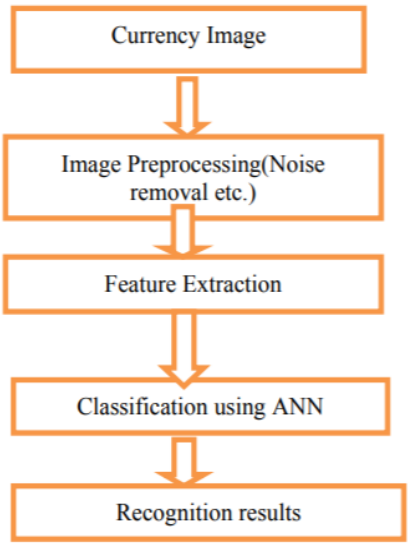

Figure 6: Currency Flowchart

Thai fast-food image classification using deep learning [20]: This system is responsible for classifying Thai fast food. A dataset of 3,960 images was used for different Thai fast food images. This dataset was divided into eleven groups. The system took inspiration from several systems like Integrated Recognition using Convolutional Networks[21] and also Accuracy improvement of Thai food image recognition[22] which used Thai fast food dataset as well. The image first enters a deep learning layer that requires a multi-layer neural network. Then it enters the convolution layer to help in recognition. Finally, it enters the classifier layer which works on the features of the image. The image passing through layers gradually becoming smaller and smaller. This system also was able to classify TFF images by using Google Framework[23] This system's accuracy of identifying Thai Fast food is 88.33%. The disadvantage of this system is that it classifies food by a complete plate as that it cannot classify rice without omelet and so on.

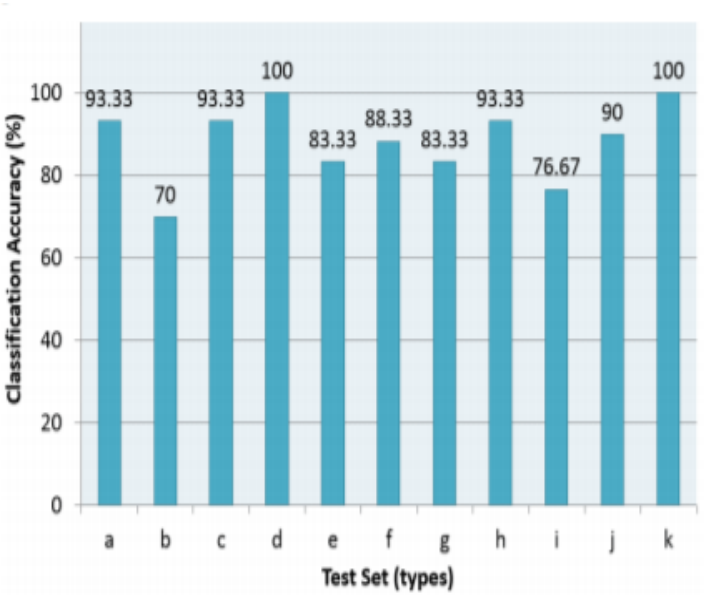

Figure 7: Prediction results of Thai fast food

2.2 Business Applications

There are some applications that target the visually impaired. Most of them target the problems that visually impaired faces like navigation and object recognition.

2.2.1 Sullivan+ (blind, visually impaired, low vision)

Sullivan application concentrated mainly on object recognition and text recognition. The user can choose what he wants to be recognized as either object, text, or face. [24]

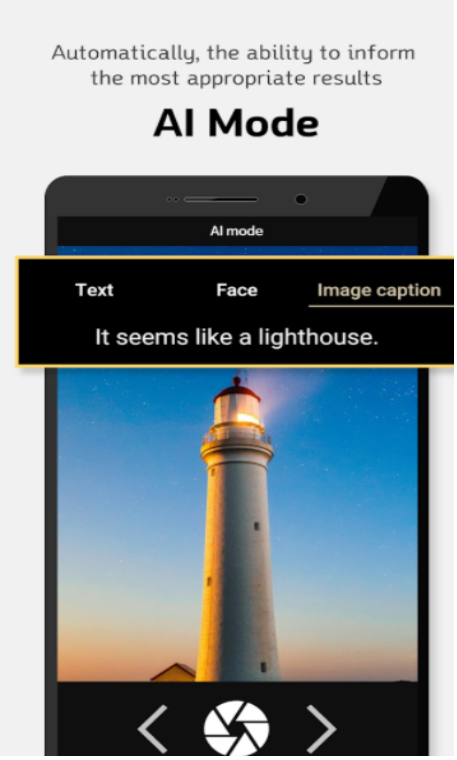

Figure 8: Object Recognition

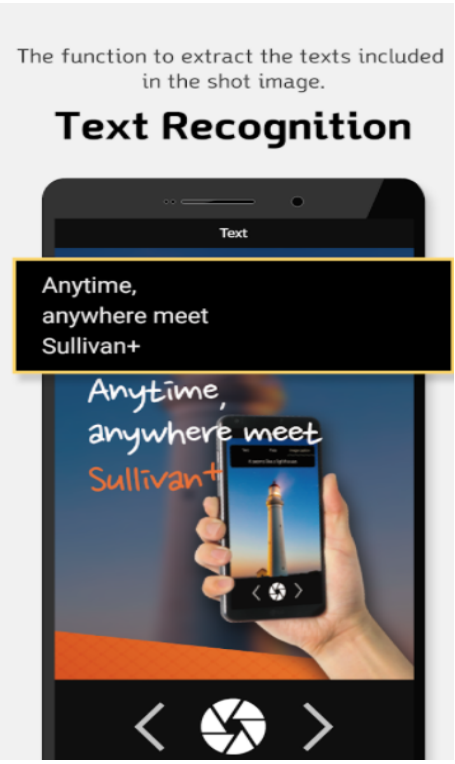

Figure 9: Text Recognition

Figure 10: Face Recognition

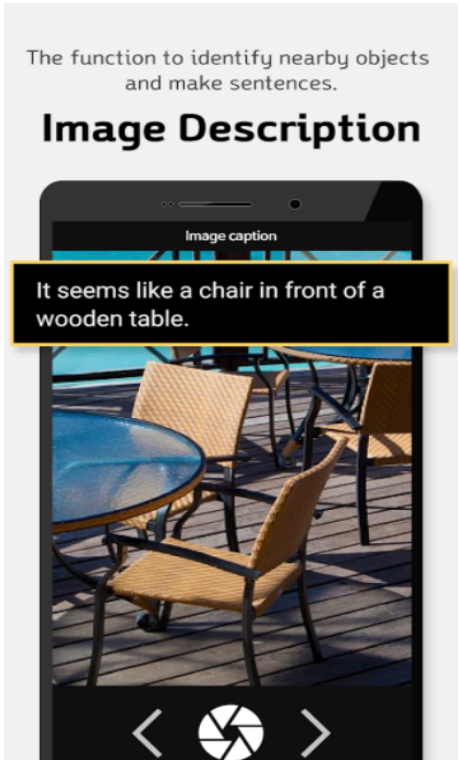

Figure 11: Image description

2.2.2 Envision AI

Envision AI application can do many different jobs in which it can convert any printed text to speech, it can identify products based on their barcodes, recognize people and their emotions, describe the color, and describe the scene surrounding the user.[25]

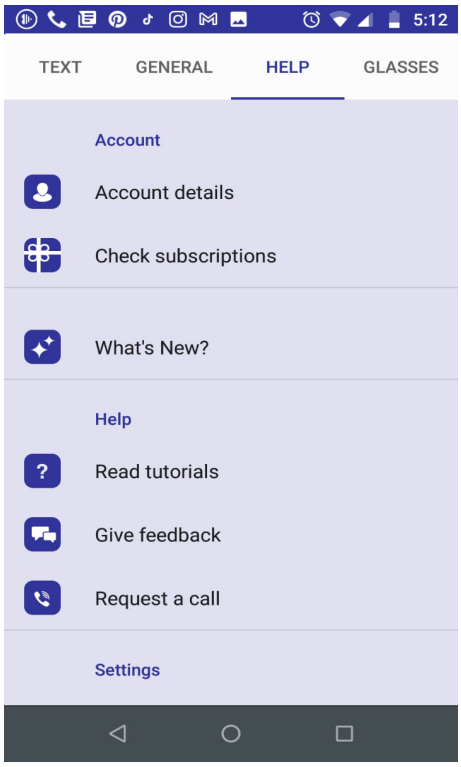

Figure 12: Envision AI application Help menu

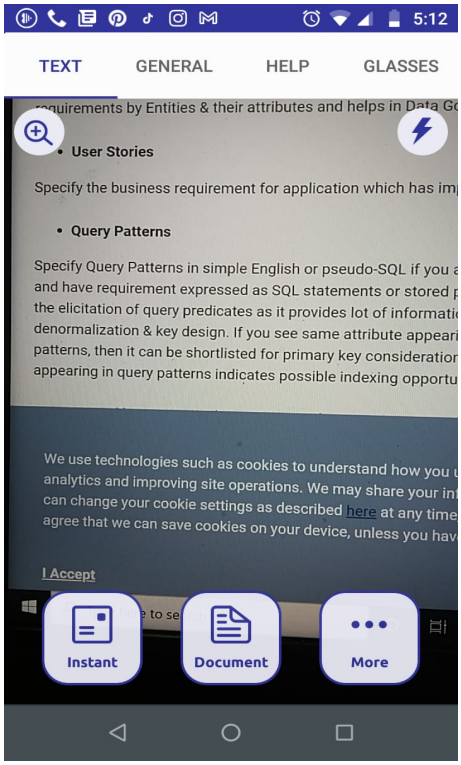

Figure 13: Envision AI application Text recognition

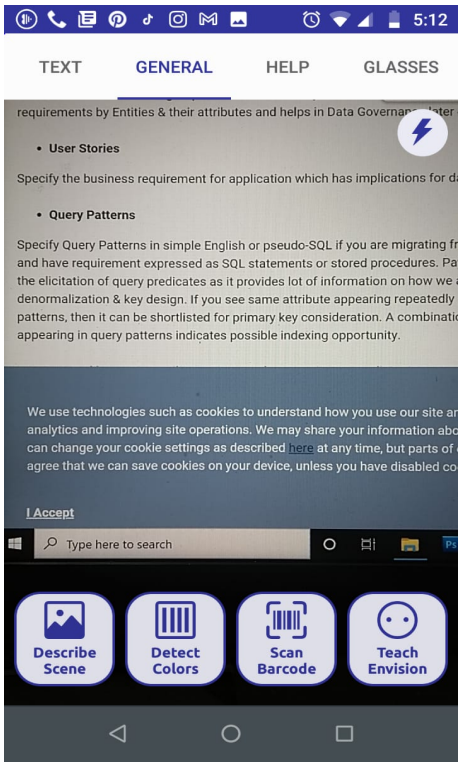

Figure 14: Envision AI application

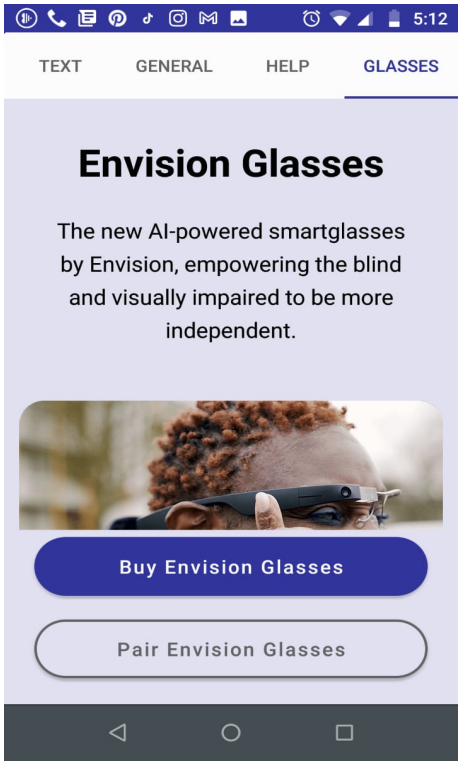

Figure 15: Envision AI application Envision Glasses

3 System Description

3.1 User Problem Statement

Visual impairment cannot allow blind people to be completely independent, so they usually depend on assistance from strangers to help them in daily activities. One of the main problems they face when they deal with the Egyptian money bills is that it does not have braille on them[26]. Braille makes it easier for visually impaired people to identify the value they hold, but on the other hand, they're more likely to experience fraud than other people. Also, the visually impaired people rely on other people to choose and coordinate their outfits because they don't know the color and pattern of the article of clothing they're planning to wear. Besides, they struggle when reading menus and printed documents. And when served food they almost always need someone else to tell them what is exactly present on their plate.

3.2 User Objectives

By using our system, visually impaired people could use it freely. The user can read text from images by the photo to speech feature, Egyptian banknotes could be identified and the amount of money the user carries. Food detection feature that identifies the type of food served on a plate and texture recognition feature that identifies color and textures.

3.3 User Characteristics

Users of the application include people who have any type of visual impairment referring to California optometric association [27] which are: people who suffer from total vision loss or who have a loss in central vision that creates a blur or a blind-spot but the person still sees, the blurred vision which causes far and near to appear to be out of the focus, night blindness or inability to see properly at night. All these mentioned impairments must use smartphones to enable them to access all the functionalities provided by the application. They also should be able to access the smartphone's camera and take pictures using it.

3.4 System Context

The diagram below shows the necessary components in the system which the visually impaired user needs.It assists the visually impaired user by giving access to the smartphone's microphone and camera, allowing the user to give commands and capture image for the user to make his/her activities.Also the user can save his/her data in the system for easier accessibility.

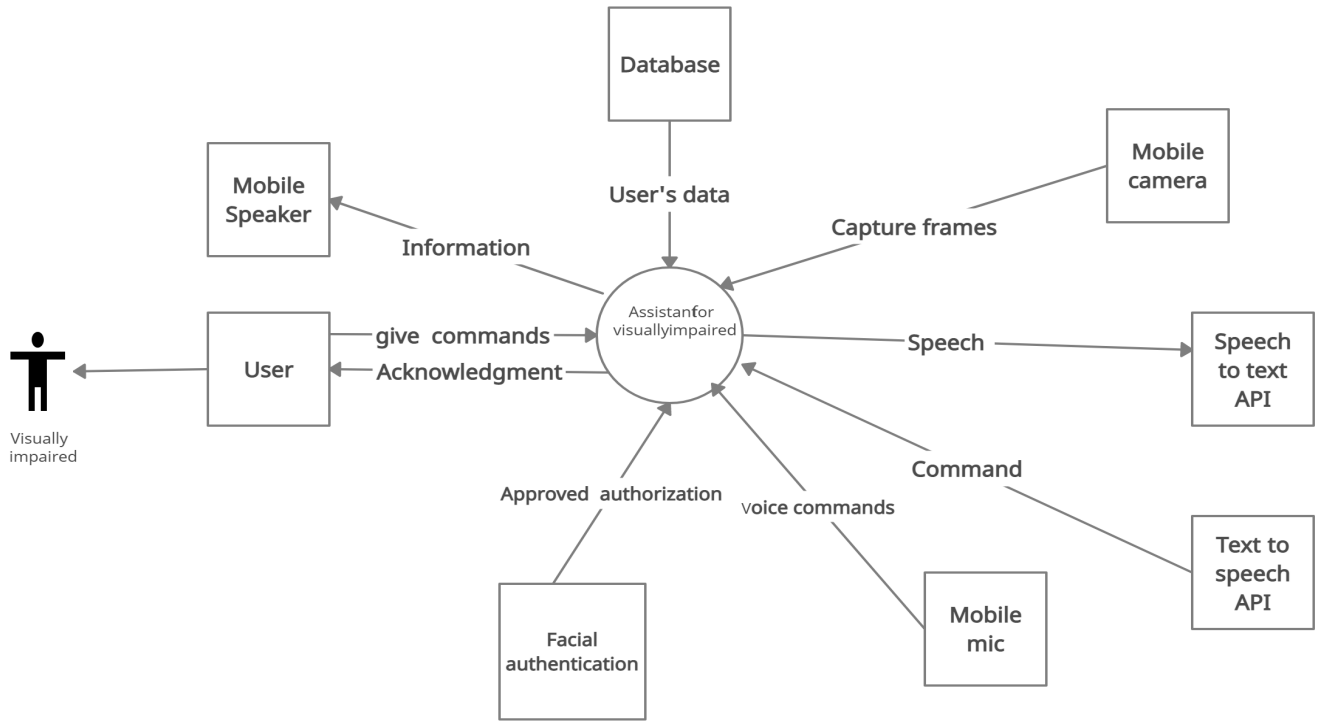

Figure 16: Visually impaired application Context diagram

4 Functional Requirements

4.1 System Functions

1. The admin shall be able to login.
2. The admin shall be able to view users' information.
3. The admin shall be able to edit users' information.
4. The admin shall be able to remove users from the system.
5. The admin shall be able to logout.
6. The user shall be able to sign up.
7. The user shall be able to login using facial recognition.
8. The user shall be able to log out.
9. The user shall be able to hear the audio menu.
10. The users shall be able to give voice commands and choose the module they want to use.
11. The user shall be able to capture an image with the help of the system.
12. The user shall be able to upload image from mobile gallery.
13. The user shall be able to hear an identification of the color and pattern of an article of cloth in the captured image.
14. The user shall be able to hear an the amount of money in the captured image.
15. The user shall be able to hear an identification of the food in the captured image.
16. The users shall be able to hear the menu in the captured image read out loud to them.

4.2 Detailed Functional Specification

Name	Registration
Code	FR01
Priority	Extreme
Critical	10/10
Description	User can register using facial recognition API to save the their facial id and enter their personal data
Input	Video stream of user's face and user's personal data
Output	Redirect to homepage
Pre-condition	None
Post-condition	User facial data and personal information is stored in database
Dependency	FR02
Risk	The function may not successfully detect the face of the user or could fail to save user facial data in database

Table 2: SignUp

Name	Facial Recognition
Code	FR02
Priority	Extreme
Critical	10/10
Description	Plugin used to identify the user existing in frame
Input	Image frame containing user's face
Output	None
Pre-condition	None
Post-condition	Aquired facial features of user
Dependency	None
Risk	The function may not successfully detect the face of the user

Table 3: Facial Recognition API

Name	User Login
Code	FR03
Priority	Extreme
Critical	10/10
Description	User can login to existing account using the facial recognition API
Input	Video stream of user's face
Output	Redirect to homepage
Pre-condition	User account must exist in database
Post-condition	Image frame captured and processed using facial recognition
Dependency	FR02
Risk	The function may wrongly redirect user to the wrong account

Table 4: User Login with facial recognition

Name	Audio Menu
Code	FR05
Priority	Extreme
Critical	9/10
Description	The user is played an audio file explaining what the system does and what to say to trigger a certain action
Input	None
Output	Audio file
Pre-condition	User must be logged in
Post-condition	Audio menu is played to the user
Dependency	FR03
Risk	none.

Table 5: Audio Menu

Name	Voice Commands
Code	FR05
Priority	Extreme
Critical	9/10
Description	The user uses the voice commands feature to trigger the camera and another certain action in the system
Input	Audio
Output	none
Pre-condition	User must be logged in and audio menu should've been played
Post-condition	Camera and another certain action are triggered
Dependency	FR03 FR05
Risk	The mic might not work probably so the speech recognition API detects a wrong word.

Table 6: Voice Commands

Name	Live Stream
Code	FR06
Priority	Extreme
Critical	10/10
Description	The Camera opens a video live stream and starts filming
Input	None
Output	Camera view
Pre-condition	User must have used the voice commands
Post-condition	Camera starts filming a video live stream
Dependency	FR05
Risk	The system may be denied access to camera.

Table 7: Camera Live Stream

Name	Object Detection
Code	FR07
Priority	Extreme
Critical	10/10
Description	Real time object detection process every frame of live stream and detects if the specified object is present in the frame
Input	Frames of the video
Output	None
Pre-condition	The camera video stream should be already on
Post-condition	Detection of specified object
Dependency	FR05 FR06
Risk	The function may not successfully detect the specified object present in the image .

Table 8: Real time object detection

Name	Generate Alert
Code	FR08
Priority	Low
Critical	5/10
Description	An Alert is generated when the specified object is detected in the live video stream so the user knows that the appropriate picture is going to be taken
Input	None
Output	Audio-able alert
Pre-condition	The specified object have been detected
Post-condition	Audio-able alert is generated.
Dependency	FR07
Risk	None

Table 9: Generate Alert for live video stream

Name	Capture Image
Code	FR09
Priority	Extreme
Critical	10/10
Description	When the alert is generated it triggers the capture method, so it can take a screenshot from the live video stream and terminates it.
Input	None
Output	Audio telling the user the appropriate image in the processing phase
Pre-condition	Alert generated to trigger the capture method
Post-condition	An appropriate screenshot is taken.
Dependency	FR07, FR08
Risk	The function may not trigger the screenshot function.

Table 10: Capture Appropriate Image

Name	Classification of banknotes
Code	FR10
Priority	Extreme
Critical	10/10
Description	Image of banknotes uploaded or taken by user undergoes classification to identify correct amount of banknote.
Input	Image
Output	Identified amount of banknote
Pre-condition	Image taken or uploaded by user
Post-condition	Redirected to screen containing output of processing
Dependency	FR10 FR11
Risk	The function may not successfully classify correct amount of banknote.

Table 11: Classification of banknotes

Name	Detection of food
Code	FR11
Priority	High
Critical	9/10
Description	Image of food is taken or uploaded by user to correctly identify food present in image with the help of AWS Rekognition API.
Input	Image
Output	Identified food
Pre-condition	Image taken or uploaded by user
Post-condition	Redirected to screen containing output of processing
Dependency	FR09
Risk	The function may not successfully detect any food present in image and could produce wrong results

Table 12: Detection of food

Name	Detection of text
Code	FR12
Priority	High
Critical	9/10
Description	Contour of text is detected to recognize the characters one by one
Input	Image
Output	Characters of text detected through contour of characters
Pre-condition	Image taken or uploaded by user
Post-condition	Redirected to screen containing classified result
Dependency	FR09
Risk	The function may not successfully detect the characters present in the image

Table 13: Detection of text

Name	Color detection in cloth
Code	FR13
Priority	Extreme
Critical	9/10
Description	Detects the color of an article of cloth according to the extracted features using indexed color histogram and SIFT methods
Input	Image of the article of clothing
Output	None
Pre-condition	The appropriate features must be extracted from the image
Post-condition	Color is detected
Dependency	FR09
Risk	The function may not successfully detect any colors present in the image and could produce wrong results

Table 14: Color Detection for cloth

Name	Pattern detection in cloth
Code	FR14
Priority	Extreme
Critical	9/10
Description	Detects the pattern present on an article of cloth according to the extracted features from the GLCM method
Input	Features extracted from the GLCM method.
Output	None
Pre-condition	The appropriate features must be extracted from the image
Post-condition	Pattern is detected
Dependency	FR09
Risk	The function may not successfully detect any patterns present in the image and could produce wrong results

Table 15: Pattern Detection for cloth

Name	Classified Result
Code	FR15
Priority	Extreme
Critical	10/10
Description	The detected colors and patterns are converted to audio using a TTS engine so the user can hear them.
Input	Detected colors and patterns
Output	Audio
Pre-condition	The colors and patterns are detected
Post-condition	Classified results are converted to audio and played to the user
Dependency	FR15 FR16
Risk	The classified patterns and colors in FR15 and FR16 may be wrong and the audio file played to the user will contain the wrong classified results.

Table 16: Classified result for cloth

Name	Admit edit
Code	FR16
Priority	High
Critical	8/10
Description	Admin can edit the data of users that is present in the system.
Input	The data that the admin will change regarding user data
Output	Conformation message of update
Pre-condition	Existing data of users
Post-condition	Updated data of users
Dependency	FR04
Risk	Queries may not be executed successfully

Table 17: Admin edit function

Name	Admin view
Code	FR17
Priority	High
Critical	8/10
Description	Admin can view all details of users present in database
Input	None
Output	None
Pre-condition	Existing data of users
Post-condition	None
Dependency	FR04
Risk	Information present in database might not be viewed

Table 18: Admin view function

Name	Admin remove
Code	FR18
Priority	High
Critical	8/10
Description	Admin can delete users present in database
Input	User id that would be removed from database
Output	Conformation message of deletion
Pre-condition	Existing data of users
Post-condition	Removal of specific user
Dependency	FR05
Risk	Delete query may not succeed

Table 19: Admin Remove function

5 Interface Requirements

5.1 User Interfaces

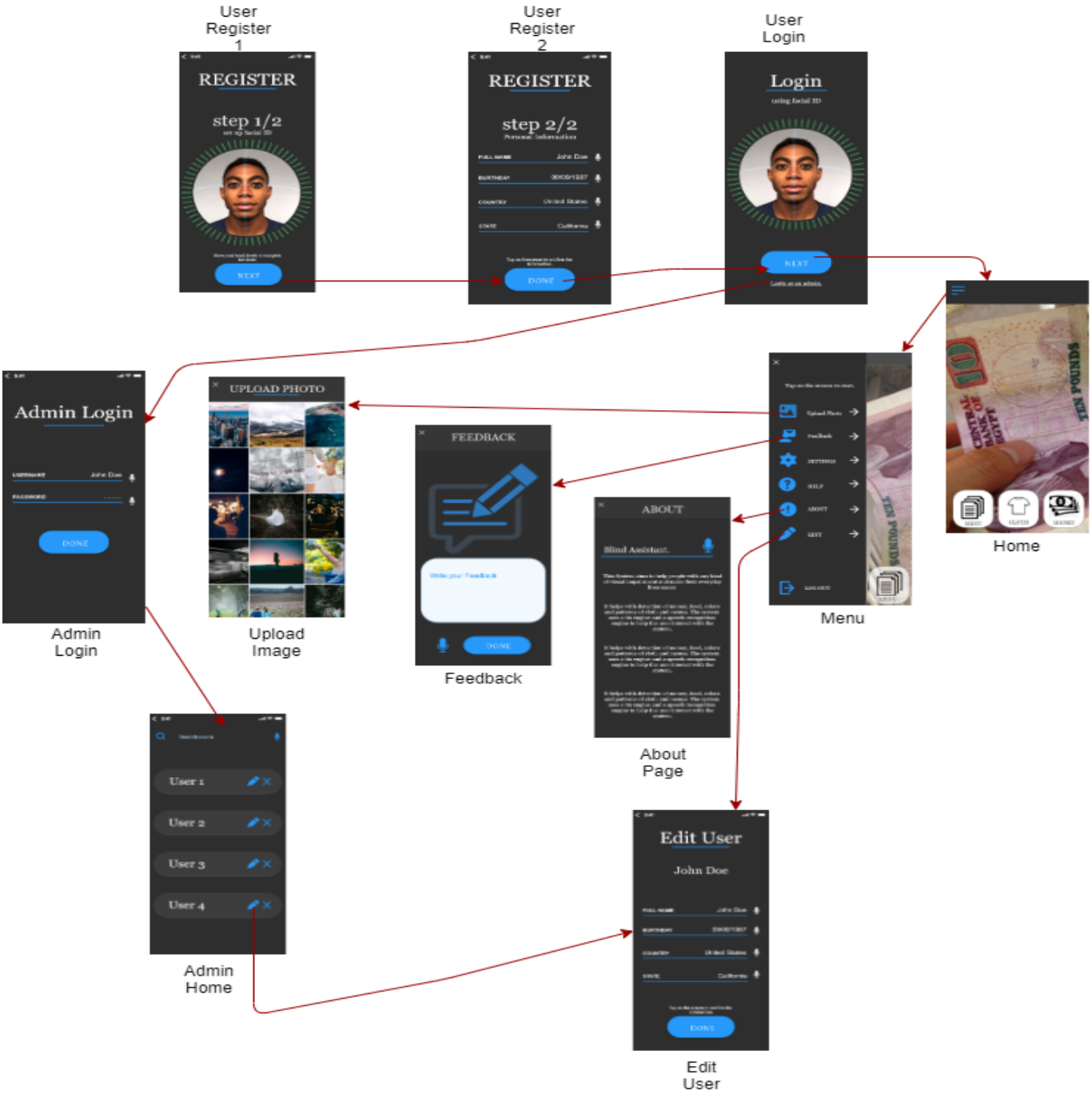

Figure 17: Wireframes

5.1.1 GUI

The user will use face recognition to login or sign-up to the system and will be able to use pre-defined voice commands that control certain features. Then the user will use the camera on the smartphone to take a picture of the item they need help with. With the voice command given, the system will process that image and execute the connected feature/algo to the command. The system has four main features photo-to-speech that reads the text out to the user, money bill recognition that identifies the Egyptian bills and the amount detected through the image, color and texture recognition that identifies the colors and textures present in the image and food recognition that identifies the food and placement of each item on the plate.

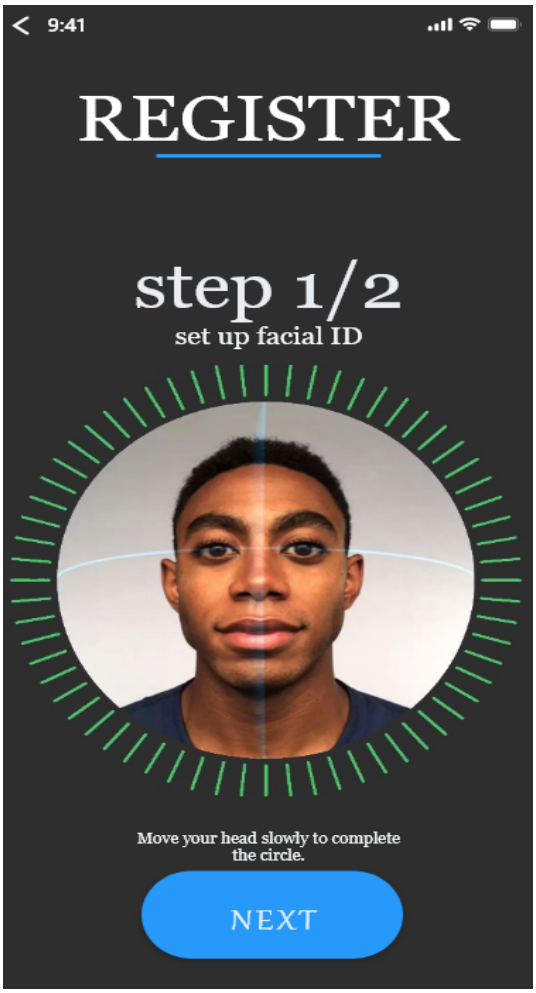

Figure 18: Setting up user's facial id

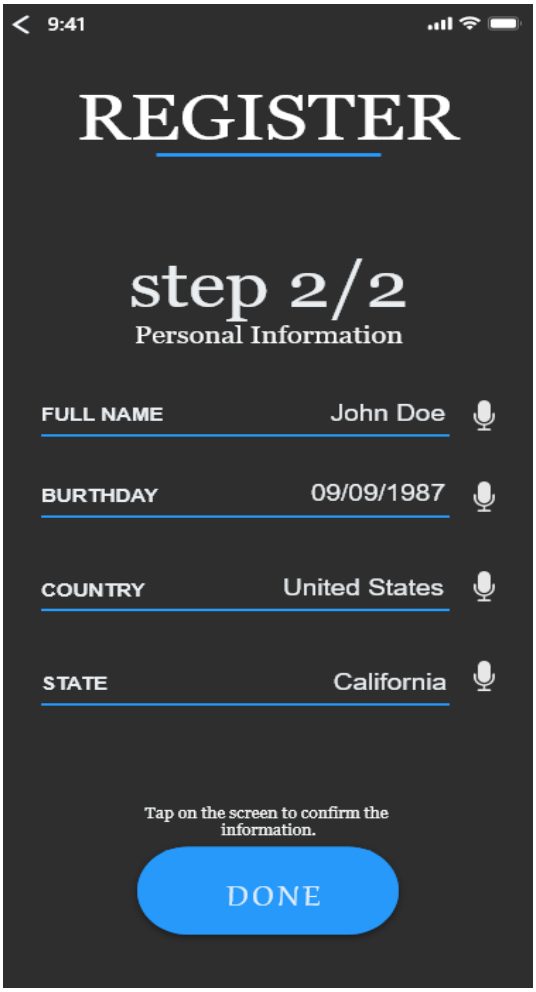

Figure 19: User's registration information

Figure 20: User Login

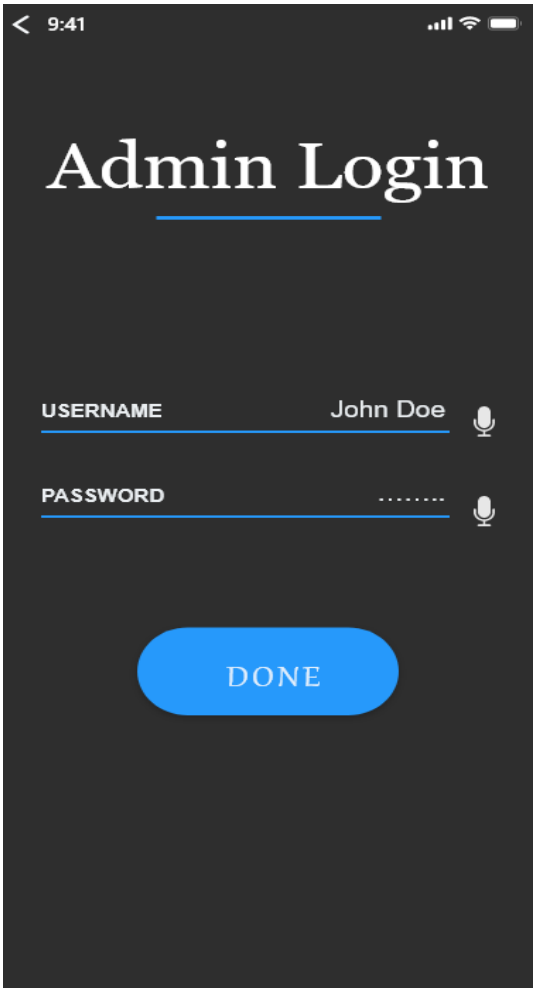

Figure 21: Admin Login

Figure 22: User Home page

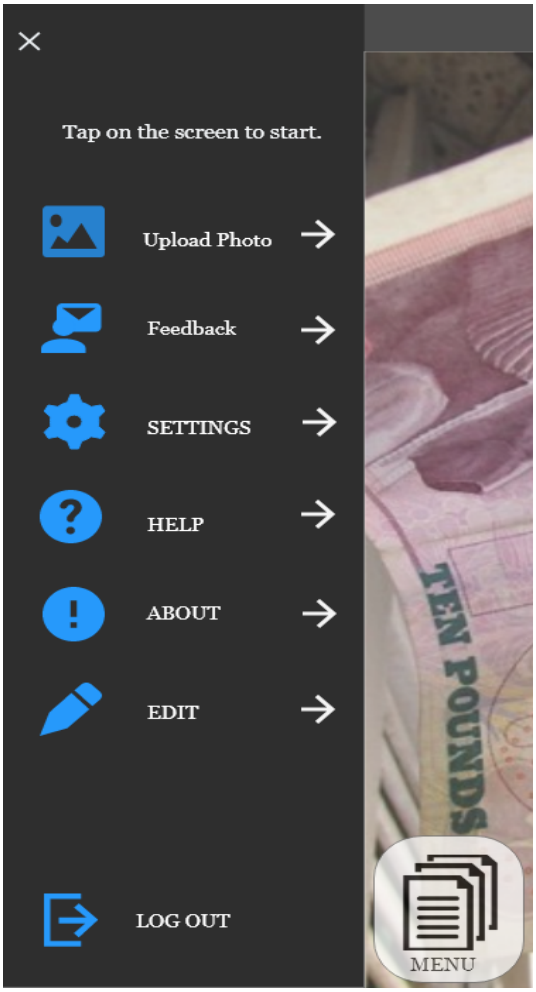

Figure 23: Menu

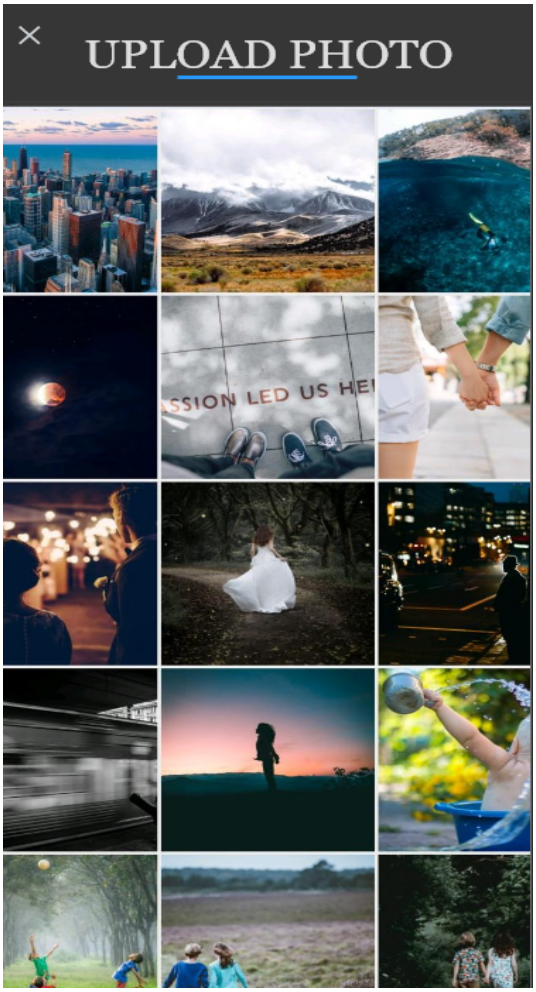

Figure 24: Upload an Image from gallery

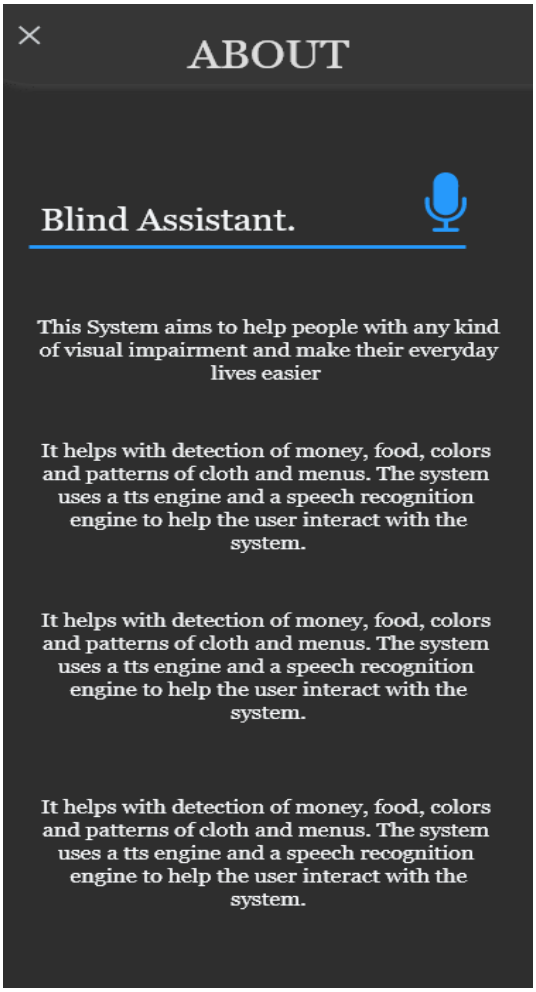

Figure 25: About page

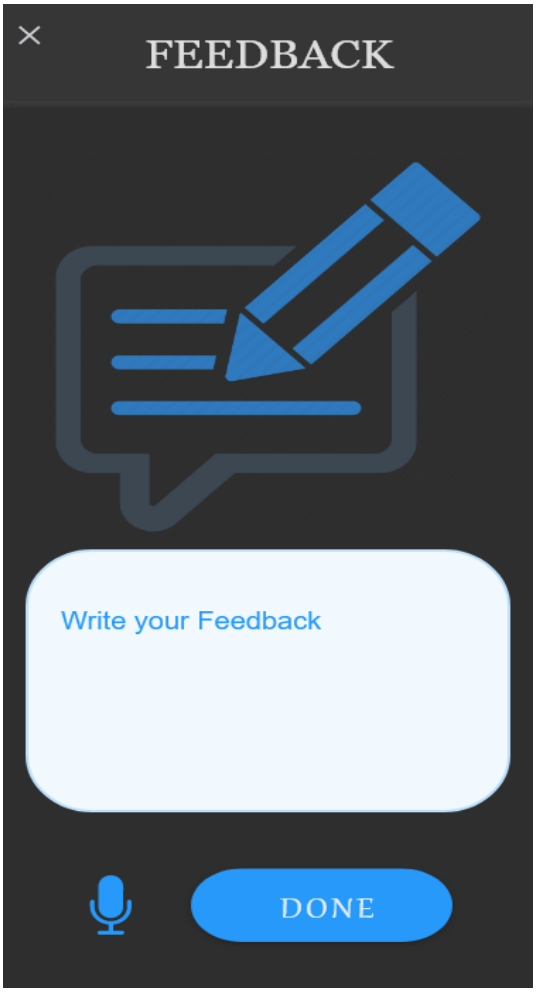

Figure 26: Feedback page

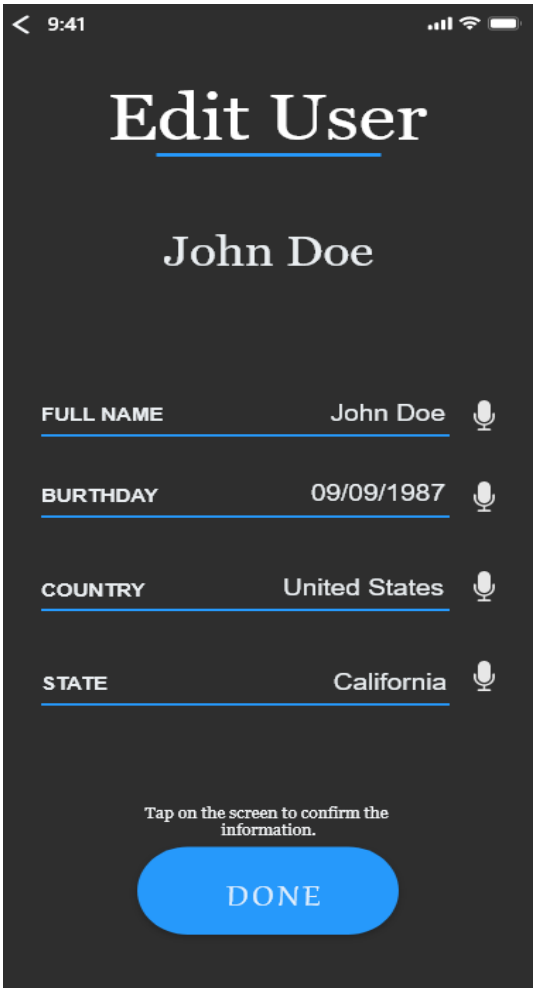

Figure 27: Edit user page

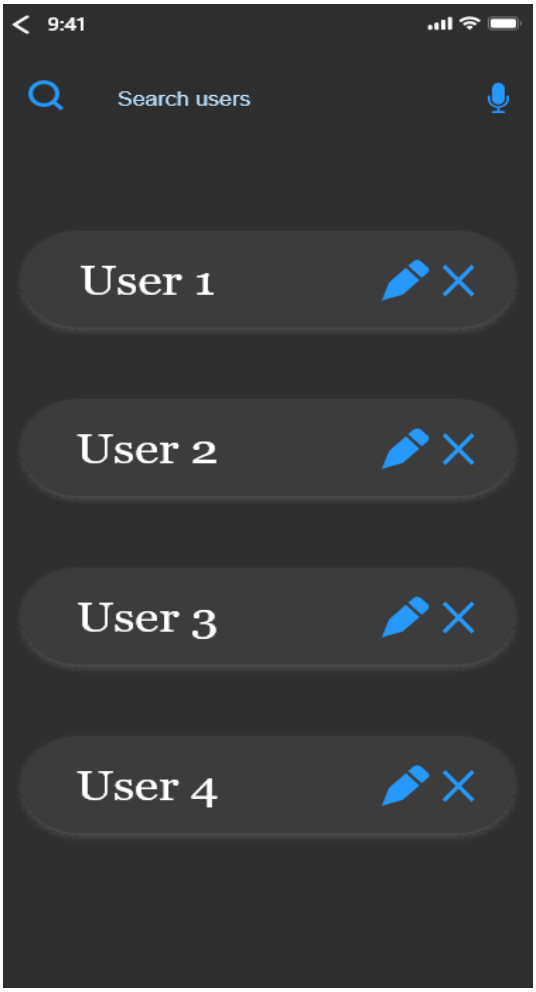

Figure 28: Admin home page

5.2 API

1. Aws rekognition API[28]: for food recognition. The API takes an image as a parameter and returns a JSON output.

2. Speech Recognition API[29]: The google speech recognition API shall be used in our implementation, where it processes the audio input that is streamed from the device's microphone and recognize the words spoken into it.
   The function used to trigger this API is called Recognizer it doesn't take any parameters. After calling the recognizer function will establish the microphone as the source of input of stream. Then we remove all the excessive noise by calling adjust-for-ambient-noise() function that takes the source as a parameter and make the recognizer listen for any input using the listen function. And then identify the word using the recognize-google() function which takes one parameter the audio input and returns text.

```
#Speech Recognition API
r=sr.Recognizer()

with sr.Microphone() as source:
    r.adjust_for_ambient_noise(source)

    print("say sth")

    audio=r.listen(source)
    try:
        word= r.recognize_google(audio)
```

Figure 29: Use Case

3. Text-To-Speech(TTS) API[30]: The google TTS we are using is gTTS Api, Which allows any text to be easily converted to speech.
   The function used to trigger this API is called gTTS which takes three parameters, the text that will be converted to speech, the language used and if the text will be slowly played or not and it returns an audio file.

```
myText="For color detetction please say colours, and For patterns detetction please say texture"

language="en"

output= gTTS(text=myText, lang=language, slow=False)

output.save("output.mp3")
os.system("start output.mp3")
```

Figure 30: gTTS code snippet

6 Design Constraints

6.1 Hardware Limitations

1. The smartphone should have a camera of at least 8 megapixels.
2. The smartphone should be connected to a stable internet connection.

7 Non-functional Requirements

7.1 Security

The application should be highly secure with the information entered when the user register in the application. It shall allow users to login or sign up using facial recognition to ensure that no one else can have access to their accounts. As well, the system data administrator shall only access the database of the user and no other user can have such access.

7.2 Reliability

The system must be reliable when performing its operations. The application should correctly detect the objects present in the image and the output from the processing should produce the most accurate result. The application shall produce audible output that correctly translates the output of the processed image to the user, and be able to correctly catch the audio commands given by the user. The database update process must roll back all related update fails because it should always be updated.

7.3 Usability

The system functionalities shall not be hard for the visually impaired to use as the system could be easily navigated. The application shall allow users to use specific voice commands to access the application's functionalities and have all the produced output be converted to audio to be heard by the user.

7.4 Maintainability

The application shall be easily maintained as the features integrated into it could be upgraded individually without the need to interrupt the overall cycle of the application. The overall performance of the application shall be recorded to allow for improvement of the performance.

7.5 Portability

The system should be accessed across all platforms. Where the users of Android or IOS smartphones can use the application and have it help them on demand.

7.6 Scailability

The system should be highly scalable as it should be easily scaled up or down without interrupting the overall performance. It should be able to support large number of users at the same time. As well, the application should allow for further integration of new features into it seamlessly.

7.7 Availability

The system should be available at any network and any mobile phone at any time.

8 Data Design

8.1 Data Description

The database is not expected to be large due to the fact that the data needed to be stored is relatively small in size. The application uses pre-trained models and makes calls for APIs that contain pre-trained models for processing the image chosen or taken by the user. Only vital information is stored like the users' personal information, as well that of the admin for the login process. The database will not store the images used in training/testing of the subsystems, but will store the result of the processed image chosen by the user. The application can be used by any number of users and an admin will have access to the application through the admin side. The data sets used are explained below.

8.1.1 Currency dataset

This data set contains the Egyptian currency in English and Arabic. The data images were collected from Egyptian paper currency and the images were taken using camera. Data augmentation was used due to the fact that the data set was small in size, where slightly modified copies of the pre-existing data was added to avoid over-fitting in training. [31]

8.1.2 Clothes dataset

- This dataset contains articles of clothing with different colors and patterns to help us identify both colors and present within a garment. It is divided into training and testing parts, where the training data set is introduced to the model during training stage and the testing dataset contains images the model did not see before to test the accuracy of it. The testing dataset contains 21300 images and the training dataset contains 62300 images. Source: Kaggle [15].
- The second dataset contains clothes with different patterns. The classes are divided into checkered, dotted, floral, solid, striped and zigzag.This is used to test the models ability to differentiate between different patterns regardless of the colors present. Source: [32].

8.1.3 Colors dataset

- This dataset trains the model to identify solid colors. It includes different shades of the solid colors. The training dataset has 80 images and the testing dataset has 90 images. Source: Kaggle [33].
- The dataset contains is divided into 10 classes of colors which are black, blue, brown, green, grey,
  orange, red, violet, white and yellow. Each class contains 25 images. Source: Kaggle [34].

8.1.4 Food images

‘This dataset is a large scale dataset for food recognition. It includes 101 types of foods used to detect and recognize the food present. Each type of food have around 1000 images, were images vary from the placement of the food on the plate and other factors that produce different images to train the model with to get best accuracy. Source: [35]

8.2 Database design description

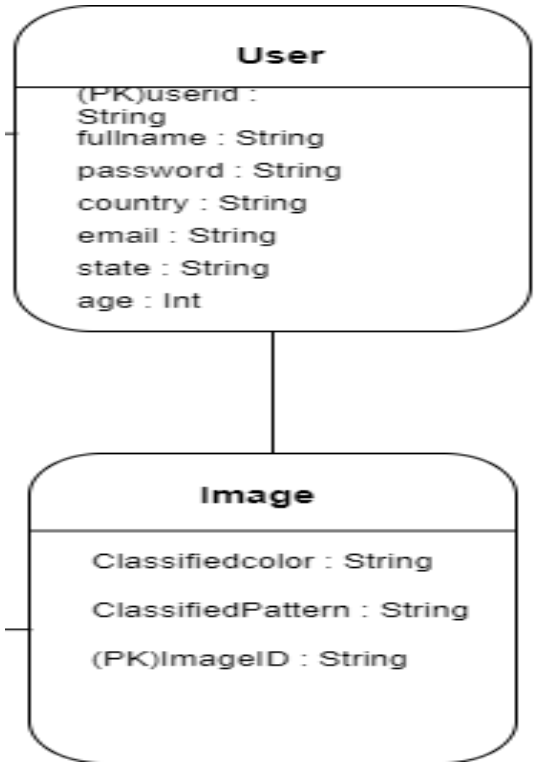

Figure 31: Database Diagram

9 Preliminary Object-Oriented Domain Analysis

9.1 Inheritance Relationships

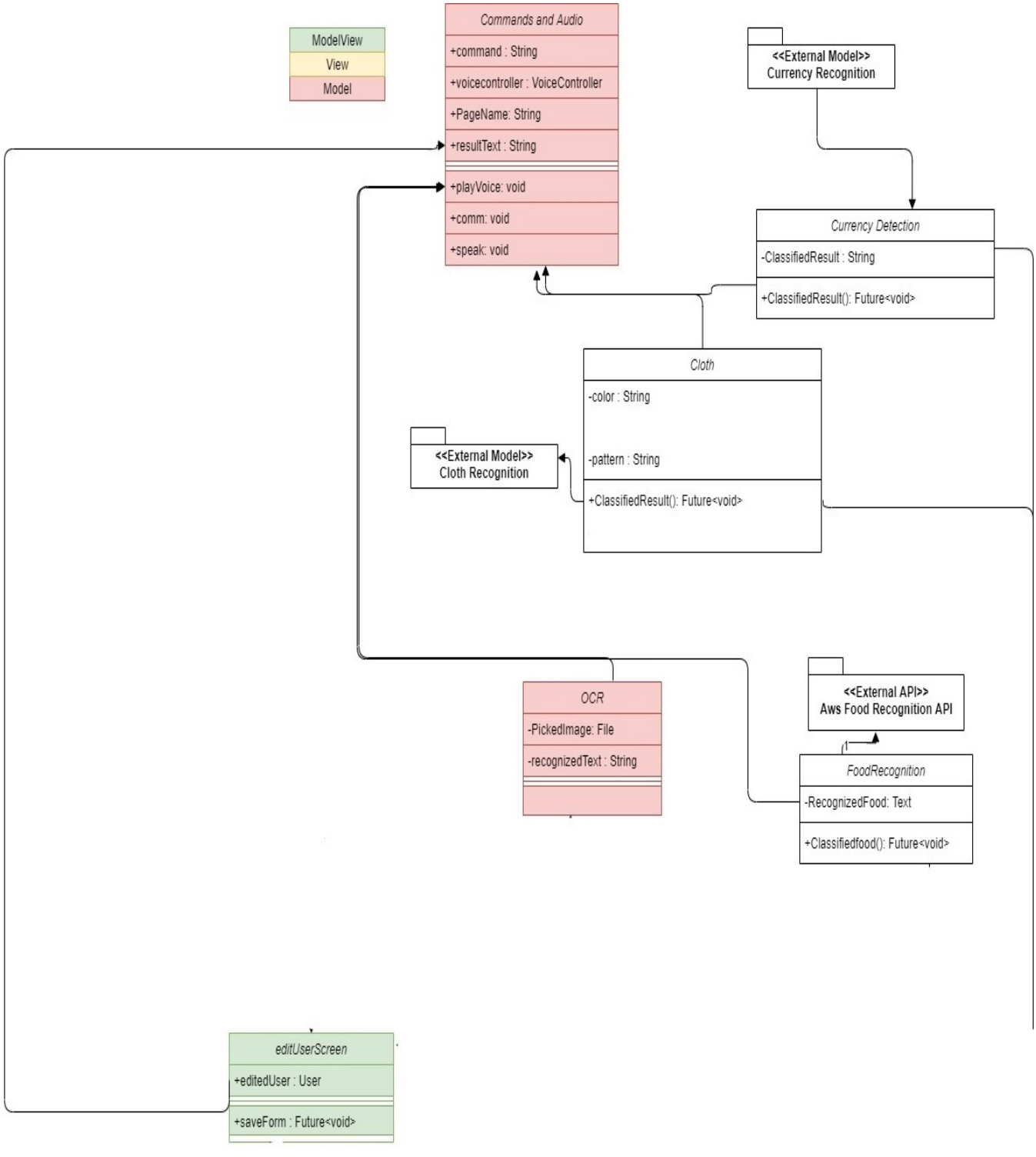

Figure 32: Inheritance of Commands And Audio class

9.2 Class descriptions

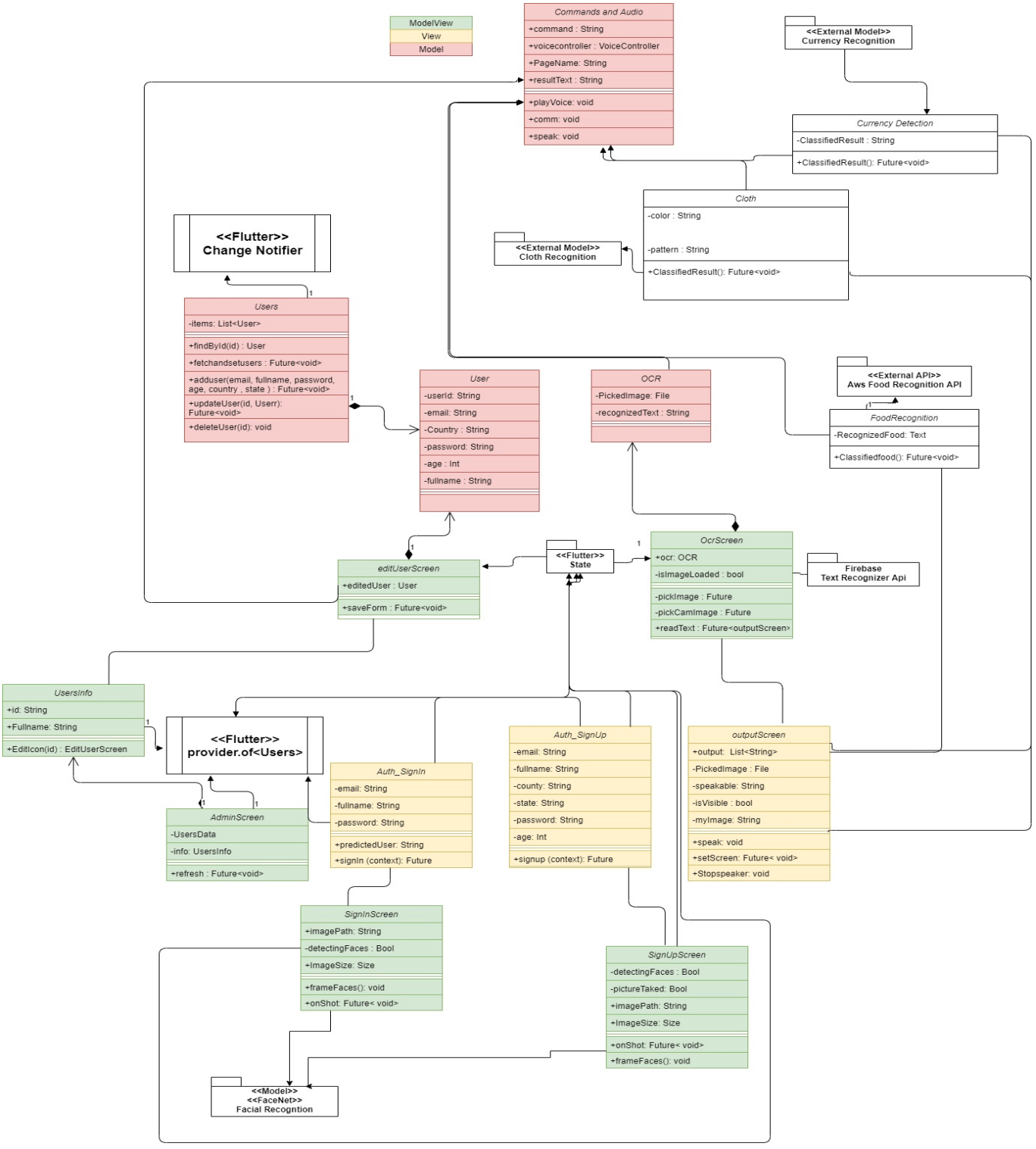

Figure 33: Class Diagram

Abstract or Concrete:	Concrete
Superclasses	None.
Subclasses	None
Purpose	Stores the users information
Collaborations	Users: Aggregation and Association. editUserScreens: Aggregation.
Attributes	userId: String fullname: String email: String password: String country: String state: String age: Int.
Operations	None

Table 20: User

Abstract or Concrete:	Concrete
Superclasses	None.
Subclasses	None
Purpose	fetch user's data from the database and edits, updates and deletes them.
Collaborations	User: Aggregation and Association.
Attributes	item: List<User>.
Operations	findById(String id): User fetchandsetusers(): Future<void> addUser(fullname, email, password, age, state, country): Future<void> UpdateUser(String id, User userr): Future<void> deleteUser(id): void.

Table 21: Users

Abstract or Concrete:	Concrete.
List of Superclasses	None.
List of Subclasses	SignupScreen. EdituserScreen. OCR. Cloth. FoodRecognition. Currency Recognetion.
Purpose	Allows the user to to give the application certain commands and hear audio files.
Collaborations	SubClasses: SignupScreen, EdituserScreen, OCR, Cloth, FoodRecognition and Currency Recognetion.
Attributes	command:String. pageName:String ResultText:String vc: VoiceController.
Operations	PlayVoice() : void. comm() : void.

Table 22: CommandsAndAudio

Abstract or Concrete:	Concrete
Superclasses	CommandsAndAudio.
Subclasses	None
Purpose	This class uses the facial recognition model so the user can register in the application.
Collaborations	CommandsAndAudio: Superclass. AuthActionButton: Association.
Attributes	detectingFaces: bool. picturedtaked: Bool imagePath: String imageSize: String
Operations	onShot(): Future<void>. frameFace():void.

Table 23: SignUpScreen

Abstract or Concrete:	Concrete
Superclasses	None.
Subclasses	None
Purpose	The admin can view all the users registered in the system.
Collaborations	UsersInfo: Aggregation .
Attributes	usersdata: UsersInfo<User>.
Operations	refresh:Future<void>.

Table 24: AdminScreen

Abstract or Concrete:	Concrete
Superclasses	None.
Subclasses	None
Purpose	Personal information of all the registered users are retrieved and send to the admin screen
Collaborations	AdminScreen: Aggregation. EditUserScreen: Association.
Attributes	id: String Fullname: String.
Operations	EditIcon(id): EditUserScreen.

Table 25: UsersInfo

Abstract or Concrete:	Concrete
Superclasses	None.
Subclasses	None
Purpose	User's personal Information is edited and saved to the database.
Collaborations	User: Aggregation .
Attributes	EditedUser: Users.
Operations	saveForm:Future<void>.

Table 26: EditUserScreen

Abstract or Concrete:	Concrete
Superclasses	None.
Subclasses	None
Purpose	This class uses the facial recognition model so the user can login.
Collaborations	AuthActionButton: Association.
Attributes	detectingFaces: bool. picturedtaked: Bool imagePath: String imageSize: String
Operations	onShot(): Future<void>. frameFace():void.

Table 27: SignInScreen

Abstract or Concrete:	Concrete
Superclasses	None.
Subclasses	None
Purpose	This class authenticates users when the log in, and where the users enter their personal information when they register so that it can be saved to the database.
Collaborations	Association: SignInScreen and SignUpScreen.
Attributes	Fullname: String. email: String password: String country: String state: String age: Int
Operations	SignUp(): Futur. SignIp(): Futur. Predicteduser():String.

Table 28: AuthActionbutton

Abstract or Concrete:	Concrete.
List of Superclasses	CommandsAndAudio.
List of Subclasses	None.
Purpose	Detectes the food in the captured image with the help of amazon's aws rekognition API.
Collaborations	outputScreen : Association. superclass: CommandsAndAudio
Attributes	RecognizedFood: String.
Operations	ClassifiedFood() : Future<void>.

Table 29: Food Recognition

Abstract or Concrete:	Concrete.
List of Superclasses	CommandsAndAudio.
List of Subclasses	None.
Purpose	Detects the text written in the captured image with the help of firebase's Text Recognizer API.
Collaborations	OcrScreen:Aggregation. superclass: CommandsAndAudio.
Attributes	PickedImage: File. recognizedText: String.
Operations	None.

Table 30: Ocr

Abstract or Concrete:	Concrete.
List of Superclasses	CommandsAndAudio.
List of Subclasses	None.
Purpose	Detects the amount of money in the captured image.
Collaborations	outputScreen : Association. CommandsAndAudio : superclass
Attributes	ClassifiedResult: String.
Operations	ClassifiedResult() : Future<void>. =

Table 31: Currency Detection

Abstract or Concrete:	Concrete.
List of Superclasses	None.
List of Subclasses	None.
Purpose	Detects if the image is loaded and extracts the text from the image.
Collaborations	outputScreen: Association. Ocr : Aggregation
Attributes	isImageloaded:Bool. ocrr:OCR
Operations	pickImage() : Future. pickCamImage() : Future.

Table 32: OcrScreen

Abstract or Concrete:	Concrete.
List of Superclasses	CommandsAndAudio.
List of Subclasses	None.
Purpose	Detects the amount of money in the captured image.
Collaborations	outputScreen : Association. CommandsAndAudio : superclass
Attributes	ClassifiedResult: String.
Operations	ClassifiedResult() : Future<void>.

Table 33: Currency Detection

Abstract or Concrete:	Concrete.
List of Superclasses	CommandsAndAudio.
List of Subclasses	None.
Purpose	Detects the color and pattern of the article of cloth that was captured in the image and suggests colors that suits the detected ones.
Collaborations	outputScreen: Association. CommandsAndAudio : superclass
Attributes	color:String. pattern:String
Operations	ClassifiedResult(color,pattern) : Future<String>.

Table 34: Cloth

Abstract or Concrete:	Concrete.
List of Superclasses	CommandsAndAudio.
List of Subclasses	None.
Purpose	Detects the amount of money in the captured image.
Collaborations	outputScreen : Association. CommandsAndAudio : superclass
Attributes	ClassifiedResult: String.
Operations	ClassifiedResult() : Future<void>.

Table 35: Currency Detection

Abstract or Concrete:	Concrete.
List of Superclasses	None.
List of Subclasses	None.
Purpose	The classified Result will be displayed to the user on this screen and converted into audio. The user will have the option to start and stop the audio.
Collaborations	Association: ocrScreen, OCR , Cloth, Currency Detection and FoodRecognition.
Attributes	speakable:String. isVisible: bool. myImage: String. PickedImage: File output: List<String>
Operations	setScreen(): Future<void>. StopSpeaker: void. speak() :void.

Table 36: OutputScreen

10 Operational Scenarios

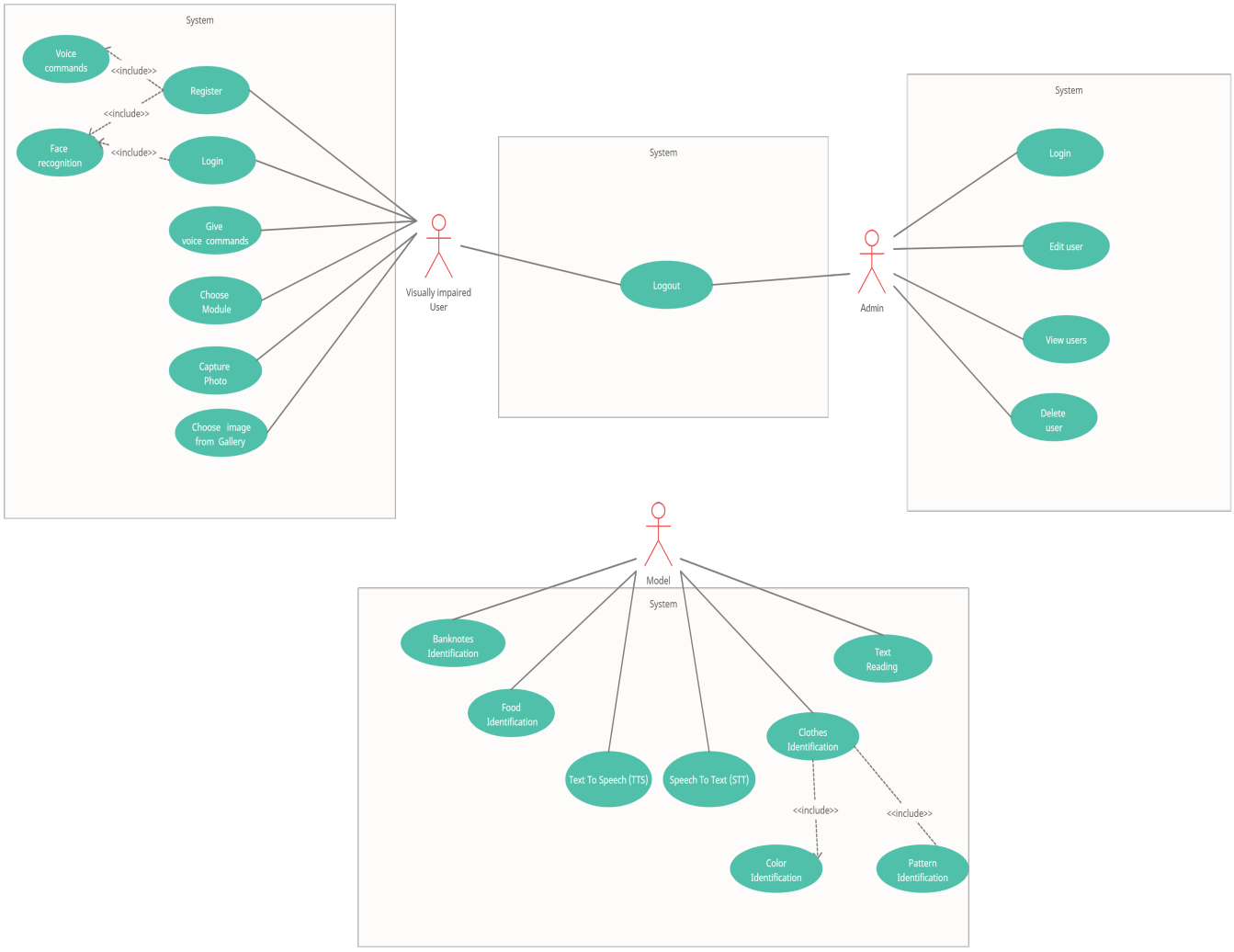

Figure 34: Use case of Visually impaired assistant application

10.1 User wants to register

A visually impaired user wants to register so he/she listens to the audio menu.Then gives the appropriate command to register.The audio menu will ask the user about his/her Full name, birthday,country and state and it will be asked one by one and the user will reply by speech and it will be converted into text.Also the camera will be activated to take the face features of the user to be able to login later on with his/her face only.Finally an audio will be played to tell the user whether the registration process succeeded or failed.If succeeded the user will be transferred to login scenario.If failed the user will be asked to repeat the registration scenario again.

10.2 User wants to login

A visually impaired user wants to login so he/she listens to the audio menu.Then gives the appropriate command to login.The camera will be activated in order to face recognition of the user.Finally an audio will be played to tell the user whether the logging in process succeeded or failed. If succeeded the user can do his/her activities.If failed the login scenario will be repeated again

10.3 User wants to know the color and pattern of an article of clothing

A user wants to know the colour and pattern of the article of clothing they chose, so he/she logs in using facial recognition, listens to the audio menu that contains the commands they could use and then gives the appropriate command to trigger the colours and patterns detection for clothes feature in the system. The user will be directed to the camera where an appropriate image will be taken through the input video stream taken by the camera. Finally the user will hear an audio of the results which is the color and the pattern of the piece of clothing.

10.4 User wants to know the amount of money they hold

A user wants to know what amount of money they have, so he/she logs in using facial recognition, listens to the audio menu that contains the commands they could use and then gives the appropriate command to trigger the currency detection feature in the system. The user will be directed to the camera where an appropriate image will be taken through the input video stream taken by the camera. Finally the user will hear an audio of the results which is the amount of money that he holds.

10.5 User wants to know the type of food

A user wants to know the type of food in the plate, so they log in using facial recognition, listens to the audio menu that contains the commands they could use and then gives the appropriate command to trigger the food detection feature in the system. where an appropriate image will be taken through the input video stream taken by the camera. Finally the user will hear an audio of the results which is the classified food that the plate contain.

10.6 User wants to know the text written on a document

A user wants to read the text in image, so they log in using facial recognition,listens to the audio menu that contains the commands they could use and then gives the appropriate command to trigger OCR feature in the system where the text will be recognized and extracted from the image. The user will be directed to the camera where an appropriate image will be taken then the user will hear an audio of text present in the image.

10.7 Admin wants to login

An admin wants to login so he/she can enter his/her email and password to login to their account. If login is successful, the admin will be redirected to their dashboard that contains all users present in the application. The admin can view the information saved about the users, edit the information or delete a user from the application. If failed the system will ask the admin to repeat the login scenario again.

10.8 Admin edits user's information

The admin wants to edit a user's information, so they are required to login first. Then choose the specific user they want to edit from the list of users. The changed data will be saved and updated in the database.

10.9 Admin removes user

The admin wants to remove a user from the system, so they required to login first. Then choose the specific user they want to remove from the list of visible users. When the confirm the removal the user will be removed from the system.

10.10 Admin view users

The admin wants to view users currently present in the system, so they are required to login first. The application will redirect to the dashboard the contains the list of users present in the system.

11 Project Plan

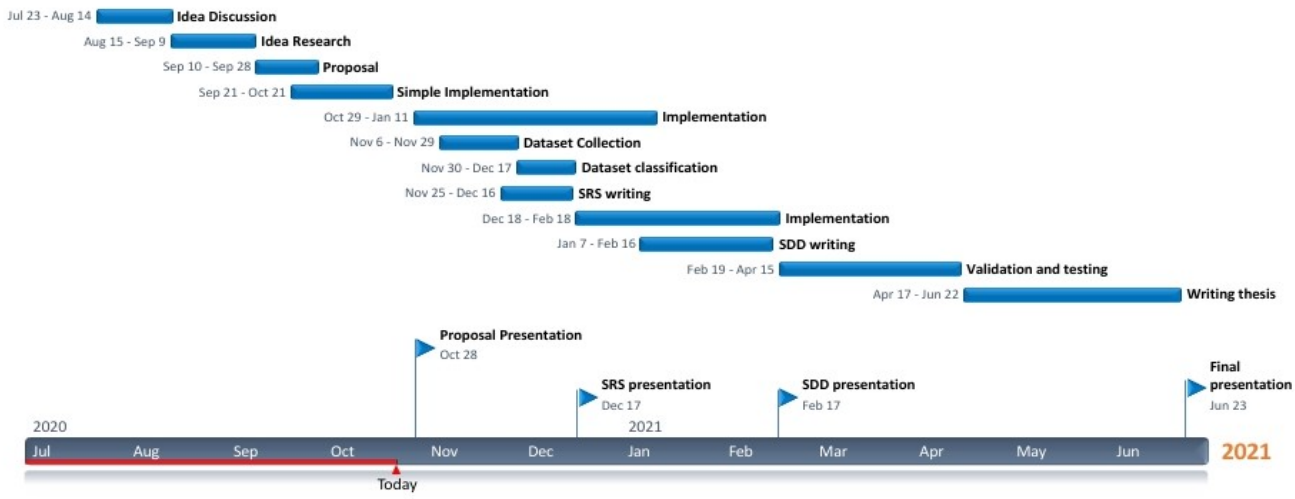

Figure 35: Project time plan

12 Appendices

12.1 Definitions, Acronyms, Abbreviations

TTS : Text to Speech
STT : Speech to Text
JSON : JavaScript Object Notation
GLCM : Gray-level Co-occurrence Matrix
OCR : Optical character recognition
GTTS : Google Text to Speech

12.2 Supportive Documents

According to the survey we made "Assistant for visually impaired", We have concluded a few points, Out of 378 responses, 329 people think assistive technology here in Egypt is too expensive to be purchased by certain social classes that could not afford such technology.

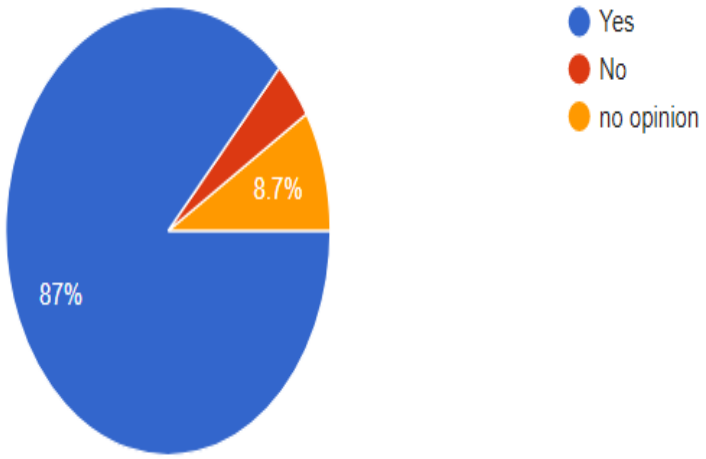

Figure 36: Statistics

Also 62.2% people who took the survey prefer if there exist two mobile applications one for free and the other with subscription payment to use the free application rather than subscription payment.

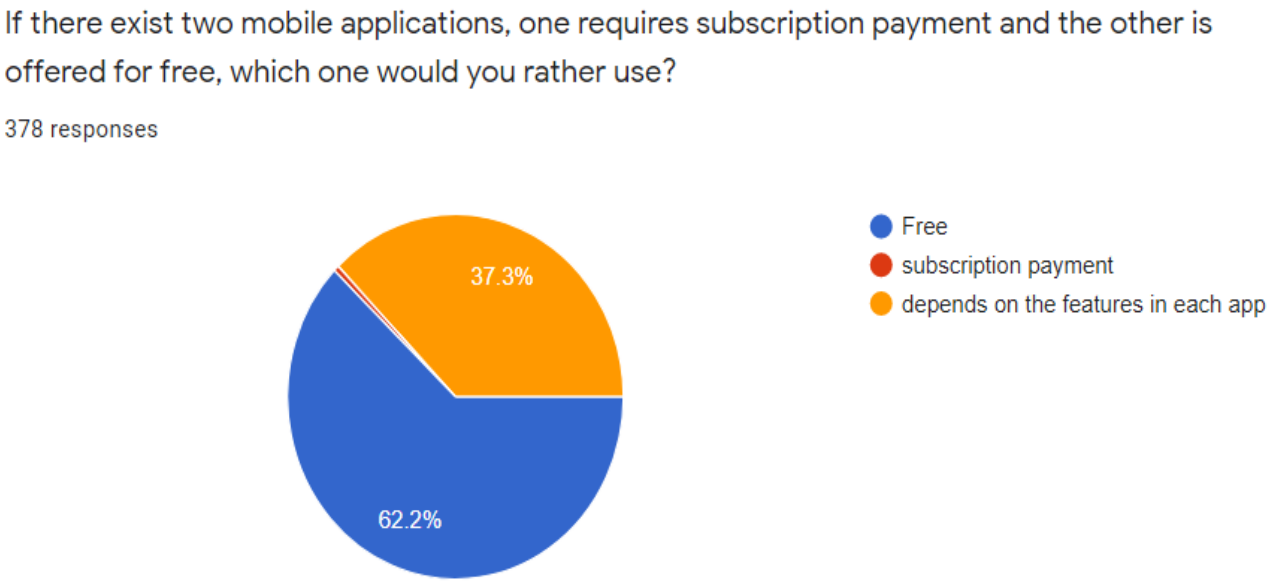

Figure 37: Statistics

We also asked our surveyors if they prefer to carry tech gadgets rather than have a mobile application that have the same features and the result was

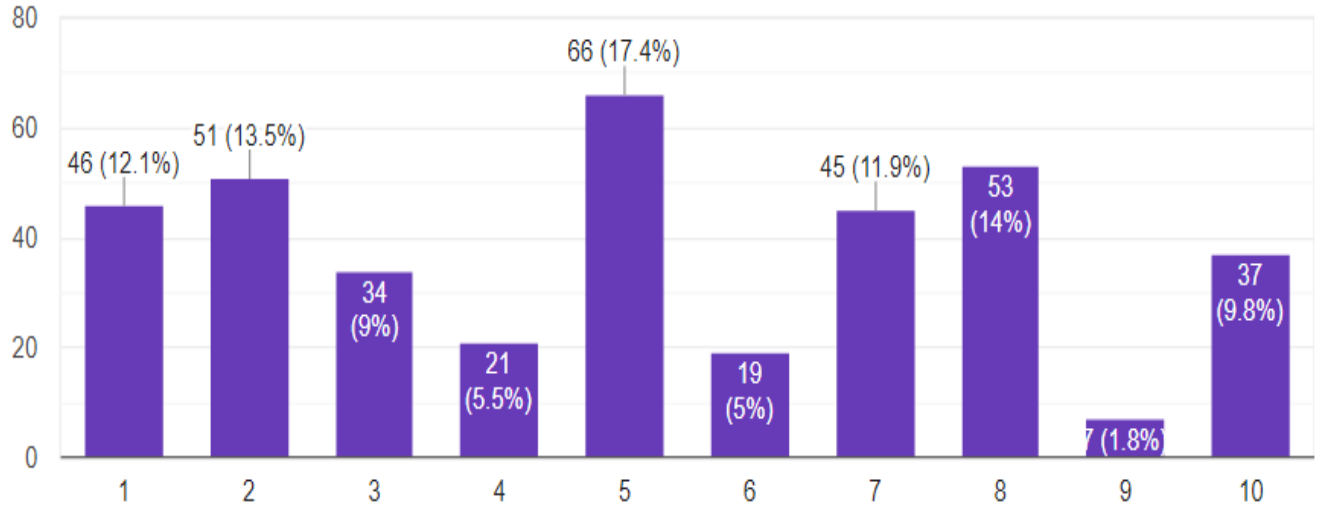

Figure 38: Statistics

References

[1]• Nico Krüger. *How to Write a Software Requirements Specification (SRS Document)*. URL: https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document.

[2] *The Importance Of Software Requirements Specification*. June 2018. URL: https://www.nascenia.com/the-importance-of-software-requirements-specification/.

[3] Shubham Melvin Felix, Sumer Kumar, and A Veeramuthu. "A smart personal AI assistant for visually impaired people". In: *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE. 2018, pp. 1245–1250.

[4] URL: https://ieeexplore.ieee.org/abstract/document/7375665.

[5] URL: https://ieeexplore.ieee.org/abstract/document/7043293.

[6] URL: https://ieeexplore.ieee.org/abstract/document/1315181.

[7] J. Wang, X. Guo, and K. Xiao. "Research and implementation of blind reader system based on android platform". In: *2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. 2016, pp. 213–216.

[8] Vaspol Ruamviboonsuk University of Washington, Vaspol Ruamviboonsuk, University of Washington, et al. *Tapulator: a non-visual calculator using natural prefix-free codes*. Oct. 2012. URL: https://dl.acm.org/doi/abs/10.1145/2384916.2384963.

[9] Frances Mary D'Andrea. *Preferences and Practices among Students who Read Braille and use Assistive Technology - Frances Mary D'Andrea, 2012*. URL: https://journals.sagepub.com/doi/abs/10.1177/0145482X1210601003.

[10] R. Ahmad, M. Z. Afzal, S. Faisal Rashid, et al. "Space Anomalies in OCRs for Arabic Like Scripts". In: *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*. 2018, pp. 67–71.

[11] J. J. Rini J. and B. Thilagavathi. "Recognizing clothes patterns and colours for blind people using neural network". In: *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. 2015, pp. 1–5.

[12] Jarbas Joaci de Mesquita Sá, André Ricardo Backes, and Paulo César Cortez. *Texture analysis and classification using shortest paths in graphs*. Apr. 2013. URL: https://www.sciencedirect.com/science/article/abs/pii/S0167865513001633.

[13] Xiaodong Yang and YingLi Tian. *Texture representations using subspace embeddings*. Mar. 2013. URL: https://www.sciencedirect.com/science/article/abs/pii/S0167865513000962.

[14] Alexander Y Ivanov, Georgii I Borzunov, and Konstantin Kogos. "Recognition and identification of the clothes in the photo or video using neural networks". In: *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. IEEE. 2018, pp. 1513–1516.

[15] Salil Gautam. *Clothes Classification*. Apr. 2018. URL: https://www.kaggle.com/salil007/caavo?select=train.

[16] S. Angsupanich and S. Matayong. "Applying the mental model for real-time recognition of Thai banknotes: The blinds' mobile application". In: *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*. 2016, pp. 86–90.

[17] M Fakrudeen, M Ali, S Yousef, et al. "Analysing the mental model of blind users in mobile touch screen devices for usability". In: *Proceedings of the World Congress on Engineering*. Vol. 2. 2013, pp. 837–842.

[18] Siti Nur Syazana Mat Saei, Suziah Sulaiman, and Halabi Hasbullah. "Mental model of blind users to assist designers in system development". In: *2010 International Symposium on Information Technology*. Vol. 1. IEEE. 2010, pp. 1–5.

[19] C. R. Jyothi, Y. K. SundaraKrishna, and V. SrinivasaRao. "Paper currency recognition for color images based on Artificial Neural Network". In: *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. 2016, pp. 3402–3406.

[20] N. Hnoohom and S. Yuenyong. "Thai fast food image classification using deep learning". In: *2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON)*. 2018, pp. 116–119. DOI: 10.1109/ECTI-NCON.2018.8378293.

[21] Pierre Sermanet, David Eigen, Xiang Zhang, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *arXiv preprint arXiv:1312.6229* (2013).

[22] Chakkrit Termritthikun and Surachet Kanprachar. "Accuracy improvement of Thai food image recognition using deep convolutional neural networks". In: *2017 international electrical engineering congress (IEECON)*. IEEE. 2017, pp. 1–4.

[23] Hamid Hassannejad, Guido Matrella, Paolo Ciampolini, et al. "Food image recognition using very deep convolutional networks". In: *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*. 2016, pp. 41–49.

[24] *Sullivan+ (blind, visually impaired, low vision) - Apps on Google Play*. URL: https://play.google.com/store/apps/details?id=tuat.kr.sullivan.

[25] *Envision AI - Apps on Google Play*. URL: https://play.google.com/store/apps/details?id=com.letsenvision.envisionai.

[26] Marie Claire Bilyk MSc, Jessica M. Sontrop PhD, PhD Gwen E. Chapman RD, et al. *Food Experiences and Eating Patterns of Visually Impaired and Blind People*. Feb. 2009. URL: https://dcjournal.ca/doi/abs/10.3148/70.1.2009.13.

[27] URL: https://www.coavision.org/m/pages.cfm?pageid=3625.

[28] Abhishek Mishra. *Machine learning in the AWS Cloud: add intellegence to applications with Amazon SageMaker and Amazon Rekognition*. 2019. URL: https://aws.amazon.com/rekognition/?blog-cards.sort-by=item.additionalFields.createdDate&blog-cards.sort-order=desc.

[29] *SpeechRecognition*. URL: https://pypi.org/project/SpeechRecognition/.

[30] *gTTS*. URL: https://pypi.org/project/gTTS/.

[31] Vedasamhitha Abburu, Saumya Gupta, S. Rimitha, et al. “Currency recognition system using image processing”. In: Aug. 2017, pp. 1–6. DOI: 10.1109/IC3.2017.8284300.

[32] Alexander J. Medeiros, Lee Stearns, Leah Findlater, et al. “Recognizing Clothing Colors and Visual Textures Using a Finger-Mounted Camera: An Initial Investigation”. In: *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS ’17*. 2017.

[33] Aydin Ayanzadeh. *color classification*. Apr. 2018. URL: https://www.kaggle.com/ayanzadeh93/color-classification.

[34] Adi Kurniawan. *Color Dataset for Color Recognition*. July 2020. URL: https://www.kaggle.com/adikurniawan/color-dataset-for-color-recognition.

[35] K Scott Mader. *Food Images (Food-101)*. May 2018. URL: https://www.kaggle.com/kmader/food41.