Menoufia University

Faculty of Computers and Information

Computer Science Department

Graduation Project Documentation

Athena

The Virtual Assistant

Supervisor: Dr. Mahmoud Hussein

July 2019

Team Members : - Muhammed Elgendi
Emad Ahmed Hatata
Mohamed Ashraf
Maged Rashed
Mahmoud Qussai
Ammar Elshaar

Supervisor :- Dr. Mahmoud Hussien
Teaching Assistant :- Eng. Ahmed Gad

ACKNOWLEDGMENT

We cannot express enough thanks to our supervisors for their continued support and encouragement: Dr. Mahmoud Hussein and Eng. Ahmed Gad. We offer our sincere appreciation for the learning opportunities provided by them.

The completion of this project could not have been accomplished without the support of our teachers, Dr. Arabi Kishk, Dr. Hamdi Mousa, and of course all our classmates.

Finally, a Big Thanks to Dr. Walid Al-Saied for the time he spent with us as he was an inspiration to us.

Table of Contents

INTRODUCTION **6**
- Overview 6
- About 6
- Similar projects 7

System Requirement Specification **8**
- Introduction 8
- Functional Requirements 8
- Non-functional Requirements 9
- System Modeling 10
  - Use Case Diagram 10
  - Sequence Diagram 12
  - Activity Diagram 14
  - Architectural Pattern 15

SYSTEM TOOLS **16**
- Mobile application 16
  - Java 16
  - Android Framework 16
  - Firebase 16
- Face Recognition 17
  - Techniques for face acquisition 17
  - Facial recognition combining different techniques 19
- Speech Recognition 19
  - Problems with Speech Recognition 20
  - How it works 20
  - Performance 23
- OCR 23
  - How optical character recognition works 23
  - Why we use it at Athena Assistant? 24
  - The implementation: 24
- Sentiment Analysis 25
- Chatbot 26

Implementation Results **29**

CONCLUSION AND FUTURE WORK **49**
- Conclusion 49
- Future Work 49

REFERENCES **50**
OCR References 50
Speech Recognition References 50
Face Recognition References 50
Android References 51
Chatbot References 51

Chapter One

1 INTRODUCTION

1.1 OVERVIEW

A decade ago, the names Alexa and Cortana were just names. These days, however, they're synonymous with intelligent or virtual assistants, which – along with the likes of Samsung Bixby, Siri, IBM Watson and Google Assistant – have redefined how we interact with our devices and software.
These “virtual butlers” can carry out a range of tasks, from making lists and scheduling appointments to ordering taxis, opening software
and activating smart devices – all controlled by simple voice commands.
Originally aimed at the consumer market, advances in artificial intelligence (AI) means that they’re becoming increasingly useful in the world of business.
“We’ve barely scratched the surface of what virtual assistants can do,” says Henrik Nordmark, head of data science at data and marketing services firm, Profusion. At the moment, they’re mostly used for basic customer service and to answer simple questions on company websites.
“But as machine learning and AI advance more broadly, it won’t be long before these assistants can interpret complex requests, predict what their owners will need, and integrate fully with all smart devices,”
he says.

1.2 ABOUT

Athena is a personal assistant with some advanced options. It can do some tasks of the personal assistant such as recognizing your voice and understanding commands like; opening some application, searching for something on the web, calling someone, setting the alarm, scheduling appointments, etc.
In addition to that, it can do some wonderful tasks such as reading computer written texts arabic and english using Optical Character Recognition (OCR) technology, recognizing faces using Face Recognition techniques and doing some sentiment analysis to those words and recognize its mood (sad, happy, angry, etc.).

1.3 Similar projects

There’re many personal assistant available and free to use online and the followings are some examples of these:

- Google Assistant
- Siri (by Apple Inc.)
- Google now
- Cortana (by Microsoft Inc.)
- Bixby (by Samsung Inc.)
- Amazon Alexa

These are only some examples and there are too many products that are available and free-to-use or pay-to-use each of which has its own advantages and disadvantages.

Chapter Two

2 System Requirement Specification

2.1 Introduction

- Project Background and Goals

The Assistant main goal is to provide the users with an easy way to add skills to the assistant. These skills that enable the user to control his own smart devices with voice commands.

- System Users (Stakeholders)

Virtual Assistant has the following stakeholder group,

- **Users**: Any user that wants to control his own devices with voice commands.

2.2 Functional Requirements

- Functional User Requirements

- The system shall enable the user to issue a command with their voice.
- The system shall enable the user to issue a command in the conversation between the application and the user.
- The system should notify the user if there are new updates available.

- Functional System Requirements

- Athena shall recognize any commands issued by the user's voice.
- Athena shall recognize any commands issued in the conversation that is between the user and the application.
- Athena should clarify to user how exactly to use a certain skill of the assistant.

2.3 Non-functional Requirements

- Athena shall keep the user's personal information safe on the user's personal phone.
- Athena should provide a simple user-friendly graphical interface to the user that is comfortable to interact with the application chatbot system and use its available skills.
- Athena should be available 24-hours with the best performance.
- Athena should not take a lot of time to respond to the user request / command even if it has lots of work to do. It should notify the user with so.
- Athena should be reliable and respond to the exact same user command or the most exact (user should not ask for something and Athena respond with a completely different thing).

2.4 System Modeling

System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system [SOMMERVILLE2010]. System modeling has generally come to mean representing the system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML). However, it is also possible to develop formal (mathematical) models of a system, usually as a detailed system specification.

2.4.1 Use Case Diagram

interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements .The actors, usually individuals involved with the system defined according to their roles also a use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors(USER).

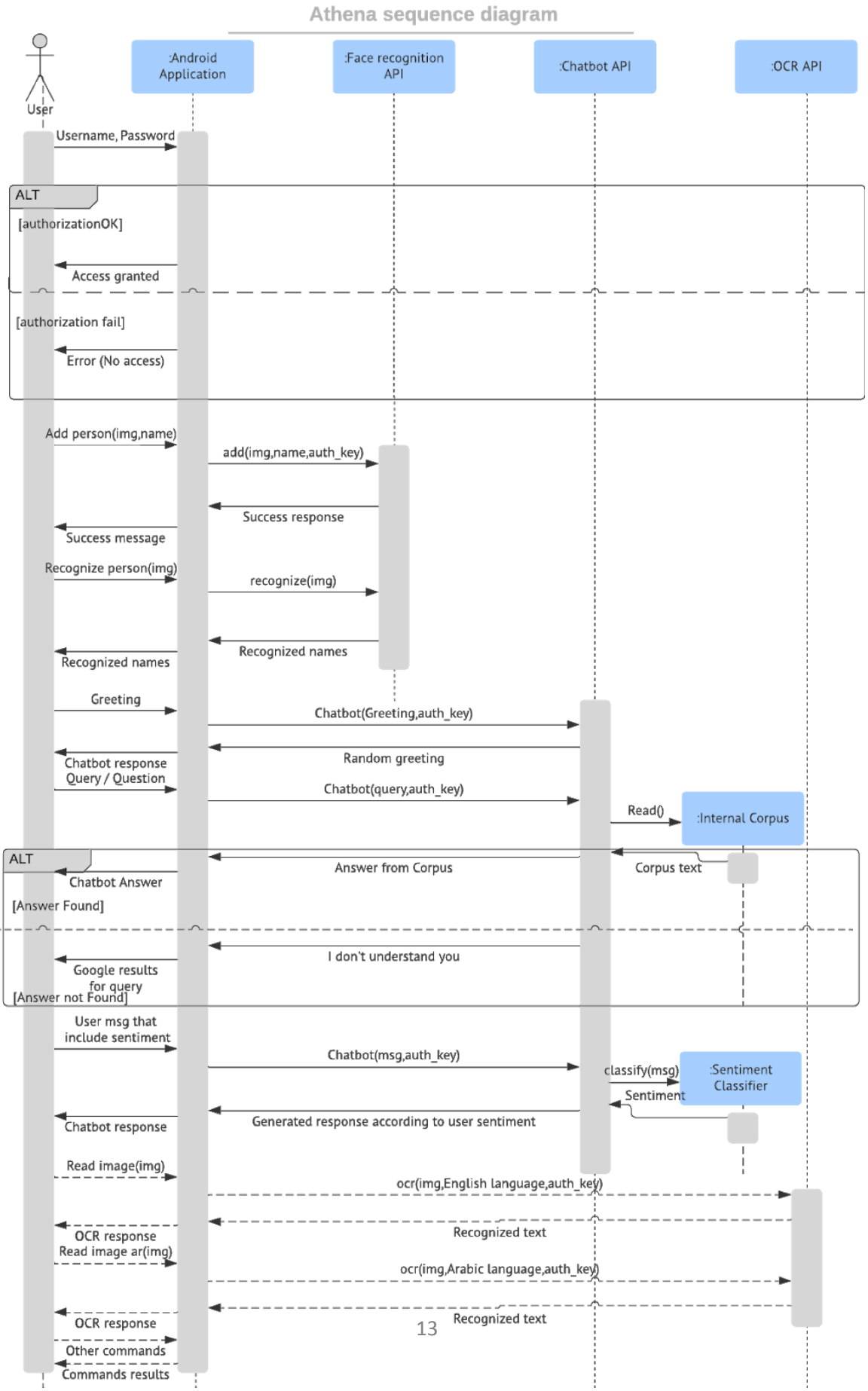

Athena use cases
System
Add person image
Issue a command to Recognize person
Issue a command to open an application
Issue a command to play music
Issue a command read an image
Send a message
Make a phone call
Open Google News
Get weather now
Add a reminder
Add an event
Show photos
Play a game
Play a video
Set alarm / timer
User

Description of Use Cases :

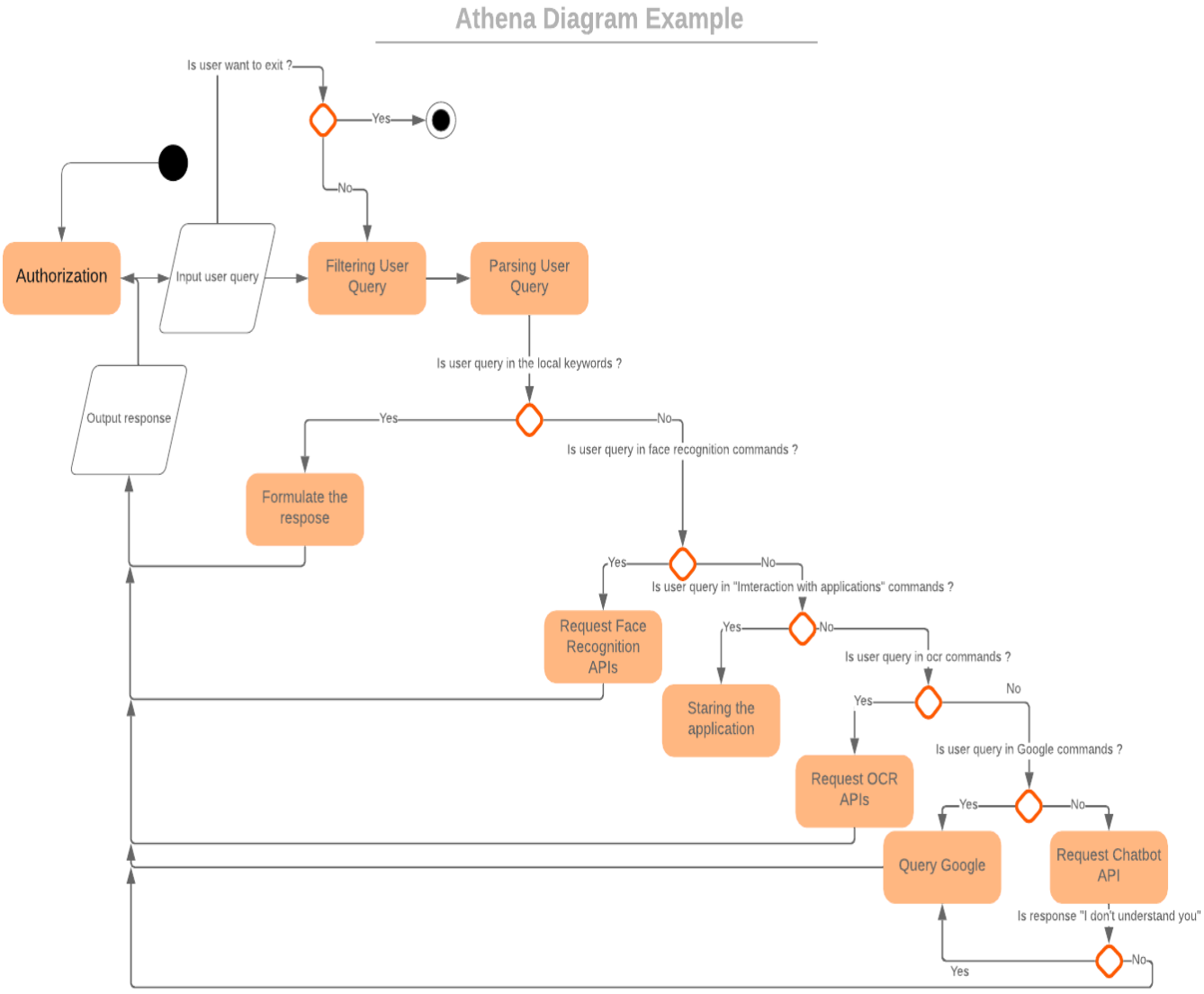
The user of the system may add a person image to the system for further recognition , issue a “open” command to open an application , issue a “play” command to play a music track, issue a “read image” command to convert computer written image to text, “Send Message” to send a text message to any phone number, “Make a phone Call“ to make a call with any contact , “ open Google News “ directs user to google trends news, “ Get Weather “ provides temperature of the city to user, “ Add a Reminder“ provides user to adding a new reminder, “Add an Event“ provides user to add new Event, “Show Photos“ just opening Gallery,
“Play a Game“ get random Game from Google Play Games, “Play Video“ Display Random Video from YouTube or any video application, “Set Timer“ enables user to set new timer.

2.4.2 Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

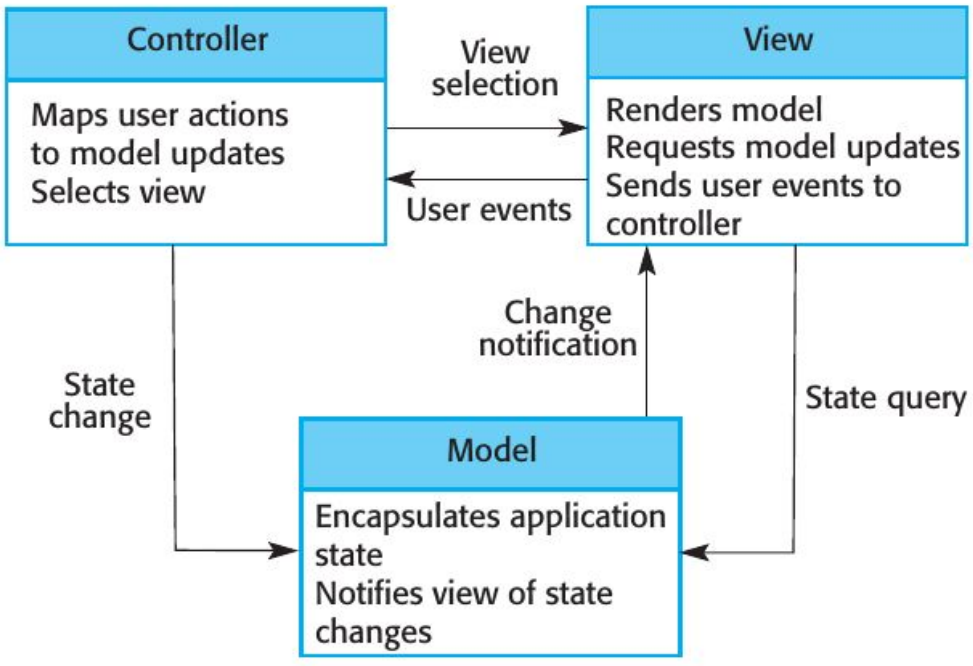

Athena sequence diagram
User
:Android Application
:Face recognition API
:Chatbot API
:OCR API
Username, Password
ALT
[authorizationOK]
Access granted
[authorization fail]
Error (No access)
Add person(img,name)
add(img,name,auth_key)
Success response
Success message
Recognize person(img)
recognize(img)
Recognized names
Recognized names
Greeting
Chatbot(Greeting,auth_key)
Random greeting
Chatbot response
Query / Question
Chatbot(query,auth_key)
Read()
:Internal Corpus
ALT
Answer from Corpus
Corpus text
Chatbot Answer
[Answer Found]
I don't understand you
Google results for query
[Answer not Found]
User msg that include sentiment
Chatbot(msg,auth_key)
classify(msg)
:Sentiment Classifier
Sentiment
Generated response according to user sentiment
Chatbot response
Read image(img)
ocr(img,English language,auth_key)
Recognized text
OCR response
Read image ar(img)
ocr(img,Arabic language,auth_key)
Recognized text
OCR response
Other commands
Commands results

2.4.3 Activity Diagram



2.4.4 Architectural Pattern

Name	MVC (Model-View-Controller)
Why used ?	1. There are multiple ways to view and interact with data (feeds fata, web pages data ,user's option data). 2. The future requirements for interaction and presentation of data are unknown because we don't know the presentation of feeds and web pages properties will change . 3. It enables us to separate presentation and manipulation of the system data so each member in the team can work independently on a specific component. 4. It enables us to structure the system into three logical component-types (Model-View-Controller).
Advantages	Allows the data to change independently of its representation and vice versa. Supports presentation of the same data in different ways, with changes made in one representation shown in all of them.
Disadvantages	May involve additional code and code complexity when the data model and interactions are simple.



Chapter Three

3 SYSTEM TOOLS

3.1 MOBILE APPLICATION

Athena simply is Virtual Assistant that gives you in multiple solutions like interacting with user applications and also capability of reading invoice or converting image to text any article even if the article in English or Arabic, also our application capable of detecting the name of person using our Face Recognition Model also to simplify the interacting user can communicate with our application using voice also using Speech Recognition and chatbot also has the capability of detecting user state even if he is sad or happy.

3.1.1 Java

We used JAVA programming language to bind the XML design with our logic like some calculations and connecting web services and dealing with libraries like Google Vision and Volley library (Google Supported) and also, we used JSoup library in parsing Google answer and more libraries.

3.1.2 Android Framework

We used in Android Framework (XML) for interacting with user and connect UI with our logic.

3.1.3 Firebase

Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase cloud.
We also used Firebase in Authentication and Authorization Phase in our Application.

3.2 Face Recognition

A Facial recognition system is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source. There are multiple methods in which facial recognition systems work, but in general, they work by comparing selected facial features from given image with faces within a database. It is also described as a Biometric Artificial Intelligence based application that can uniquely identify a person by analyzing patterns based on the person's facial textures and shape. While initially a form of computer application, it has seen wider uses in recent times on mobile platforms and in other forms of technology, such as robotics. It is typically used as access control in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems. Although the accuracy of facial recognition system as a biometric technology is lower than iris recognition and fingerprint recognition, it is widely adopted due to its contactless and non-invasive process. Recently, it has also become popular as a commercial identification and marketing tool. Other applications include advanced human-computer interaction, video surveillance, automatic indexing of images, and video database, among others.

3.2.1 Techniques for face acquisition

Essentially, the process of face recognition is performed in two steps. The first involves feature extraction and selection and, the second is the classification of objects. Later developments introduced varying technologies to the procedure. Some of the most notable include the following techniques:

- **Traditional**

Some face recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features.

Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data. One of the earliest successful systems is based on template matching techniques applied to a set of salient facial features, providing a sort of compressed face representation.

Recognition algorithms can be divided into two main approaches, geometric, which looks at distinguishing features, or photometric, which is a statistical approach

that distils an image into values and compares the values with templates to eliminate variances. Some classify these algorithms into two broad categories: holistic and feature-based models. The former attempts to recognize the face in its entirety while the feature-based subdivide into components such as according to features and analyze each as well as its spatial location with respect to other features.

Popular recognition algorithms include principal component analysis using eigenfaces, linear discriminant analysis, elastic bunch graph matching using the Fisher face algorithm, the hidden Markov model, the multilinear subspace learning using tensor representation, and the neuronal motivated dynamic link matching.

- **3-Dimensional Recognition**

Three-dimensional face recognition technique uses 3D sensors to capture information about the shape of a face. This information is then used to identify distinctive features on the surface of a face, such as the contour of the eye sockets, nose, and chin.

One advantage of 3D face recognition is that it is not affected by changes in lighting like other techniques. It can also identify a face from a range of viewing angles, including a profile view. Three-dimensional data points from a face vastly improve the precision of face recognition. 3D research is enhanced by the development of sophisticated sensors that do a better job of capturing 3D face imagery. The sensors work by projecting structured light onto the face. Up to a dozen or more of these image sensors can be placed on the same CMOS chip—each sensor captures a different part of the spectrum

Even a perfect 3D matching technique could be sensitive to expressions. For that goal a group at the Technion applied tools from metric geometry to treat expressions as isometries.

A new method is to introduce a way to capture a 3D picture by using three tracking cameras that point at different angles; one camera will be pointing at the front of the subject, the second one to the side, and the third one at an angle. All these cameras will work together so it can track a subject's face in real time and be able to face detect and recognize.

- **Skin texture analysis**

Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. This technique, called Skin Texture Analysis, turns the unique lines, patterns, and spots apparent in a person's skin into a mathematical space.

Surface Texture Analysis works much the same way facial recognition does. A picture is taken of a patch of skin, called a skin print. That patch is then broken up into smaller blocks. Using algorithms to turn the patch into a mathematical, measurable space, the system will then distinguish any lines, pores and the actual skin texture. It can identify the contrast between identical pairs, which are not yet possible using facial recognition software alone.

Tests have shown that with the addition of skin texture analysis, performance in recognizing faces can increase 20 to 25 percent.

3.2.2 Facial recognition combining different techniques

As every method has its advantages and disadvantages, technology companies have amalgamated the traditional, 3D recognition and Skin Textual Analysis, to create recognition systems that have higher rates of success.

Combined techniques have an advantage over other systems. It is relatively insensitive to changes in expression, including blinking, frowning or smiling and has the ability to compensate for mustache or beard growth and the appearance of eyeglasses. The system is also uniform with respect to race and gender.

3.3 Speech Recognition

Speech recognition refers to the process of recognizing and understanding spoken language. Input comes in the form of audio data, and the speech recognizers will process this data to extract meaningful information from it. This has a lot of practical uses, such as voice-controlled devices, transcription of spoken language into words, security systems, and so on.

Speech signals are very versatile in nature. There are many variations of speech in the same language. There are different elements to speech, such as language, emotion, tone, noise, accent, and so on. It's difficult to rigidly define a set of rules that can constitute speech. Even with all these variations, humans are really good at

understanding all of this with relative ease. Hence, we need machines to understand speech in the same way.
Over the last couple of decades, researchers have worked on various aspects of speech, such as identifying the speaker, understanding words, recognizing accents, translating speech, and so on. Among all these tasks, automatic speech recognition has been the focal point of attention for many researchers.

3.3.1 Problems with Speech Recognition

Speech recognition, in its version known as Speech to Text (STT), has also been used for a long time to translate spoken words into text. “You talk, it types,” as Via-Voice would say on its box. But there is one problem with STT as we know it. Via-Voice is one of the best in the industry, so imagine the rest. The technology has matured and improved, but speech to text still makes people ask questions. One of its main difficulties is the immense variations among people in pronouncing words.
Not all languages are supposed in speech recognition, and those that do are often not supported as well as English. As a result, most devices that run speech recognition software perform reasonably only in English.
A set of hardware requirements makes speech recognition difficult to deploy in certain cases. You need a microphone that is intelligent enough to filter off background noise but at the same time powerful enough to capture voice naturally.
Speaking of **background n**oise, it can cause a whole system to fail. As a result, speech recognition fails in many cases due to noises that are out of the user's control.
Speech recognition is proving to be better off as an input method for new phones and communication technologies like VoIP than as a productivity tool for mass text input.

3.3.2 How it works

Speech recognition works using algorithms through acoustic and language modeling. Acoustic modeling represents the relationship between linguistic units of speech and audio signals; language modeling matches sounds with word sequences to help distinguish between words that sound similar.
Often, Hidden Markov Models (HMMs) are used as well to recognize temporal patterns in speech to improve accuracy within the system.

- **Hidden Markov Model (HMM)**
  Modern general-purpose speech recognition systems are based on Hidden Markov Models. These are statistical models that output a sequence of symbols

or quantities. HMMs are used in speech recognition because a speech signal can be viewed as a piecewise stationary signal or a short-time stationary signal. In a short time-scale (e.g., 10 milliseconds), speech can be approximated as a stationary process. Speech can be thought of as a Markov model for many stochastic purposes.

Another reason why HMMs are popular is because they can be trained automatically and are simple and computationally feasible to use. In speech recognition, the hidden Markov model would output a sequence of n-dimensional real-valued vectors (with n being a small integer, such as 10), outputting one of every 10 milliseconds. The vectors would consist of cepstral coefficients, which are obtained by taking a Fourier transform of a short time window of speech and decorrelating the spectrum using a cosine transform, then taking the first (most significant) coefficients. The hidden Markov model will tend to have in each state a statistical distribution that is a mixture of diagonal covariance Gaussians, which will give a likelihood for each observed vector. Each word, or (for more general speech recognition systems), each phoneme, will have a different output distribution; a hidden Markov model for a sequence of words or phonemes is made by concatenating the individual trained hidden Markov models for the separate words and phonemes.

- **Dynamic time warping (DTW)-based speech recognition**

Dynamic time warping is an approach that was historically used for speech recognition but has now largely been displaced by the more successful HMM-based approach.

Dynamic time warping is an algorithm for measuring similarity between two sequences that may vary in time or speed. For instance, similarities in walking patterns would be detected, even if in one video the person was walking slowly and if in another he or she were walking more quickly, or even if there were accelerations and deceleration during the course of one observation. DTW has been applied to video, audio, and graphics – indeed, any data that can be turned into a linear representation can be analyzed with DTW.

A well-known application has been automatic speech recognition, to cope with different speaking speeds. In general, it is a method that allows a computer to find an optimal match between two given sequences (e.g., time series) with certain restrictions. That is, the sequences are "warped" non-linearly to match each other. This sequence alignment method is often used in the context of hidden Markov models.

- **Neural networks**

Neural networks emerged as an attractive acoustic modeling approach in ASR in the late 1980s. Since then, neural networks have been used in many aspects of speech recognition such as phoneme classification, isolated word recognition, audiovisual speech recognition, audiovisual speaker recognition and speaker adaptation.

neural networks make fewer explicit assumptions about feature statistical properties than HMMs and have several qualities making them attractive recognition models for speech recognition. When used to estimate the probabilities of a speech feature segment, neural networks allow discriminative training in a natural and efficient manner. However, in spite of their effectiveness in classifying short-time units such as individual phonemes and isolated words, early neural networks were rarely successful for continuous recognition tasks because of their limited ability to model temporal dependencies.

One approach to this limitation was to use neural networks as a pre-processing, feature transformation or dimensionality reduction, step prior to HMM based recognition. However, more recently, LSTM and related recurrent neural networks (RNNs) and Time Delay Neural Networks (TDNN's) have demonstrated improved performance in this area.

- **Deep feedforward and recurrent neural networks**

Deep Neural Networks and Denoising Autoencoders are also under investigation. A deep feedforward neural network (DNN) is an artificial neural network with multiple hidden layers of units between the input and output layers. Similar to shallow neural networks, DNNs can model complex non-linear relationships. DNN architectures generate compositional models, where extra layers enable composition of features from lower layers, giving a huge learning capacity and thus the potential of modeling complex patterns of speech data.

A success of DNNs in large vocabulary speech recognition occurred in 2010 by industrial researchers, in collaboration with academic researchers, where large output layers of the DNN based on context dependent HMM states constructed by decision trees were adopted. See comprehensive reviews of this development and of the state of the art as of October 2014 in the recent Springer book from Microsoft Research. See also the related background of automatic speech recognition and the impact of various machine learning paradigms, notably including deep learning, in recent overview articles.

One fundamental principle of deep learning is to do away with hand-crafted feature engineering and to use raw features. This principle was first explored successfully in the architecture of deep autoencoder on the "raw"

spectrogram or linear filter-bank features, showing its superiority over the Mel-Cepstral features which contain a few stages of fixed transformation from spectrograms. The true "raw" features of speech, waveforms, have more recently been shown to produce excellent larger-scale speech recognition results.

3.3.3 Performance

Speech recognition performance is measured by accuracy and speed. Accuracy is measured with Word Error Rate (WER). WER works at the word level and identifies inaccuracies in transcription, although it cannot identify how the error occurred. Speed is measured with the real-time factor. A variety of factors can affect computer speech recognition performance, including pronunciation, accent, pitch, volume and background noise.

3.4 OCR

OCR is the use of technology to distinguish printed or handwritten text characters inside digital images of physical documents, such as a scanned paper document. The basic process of OCR involves examining the text of a document and translating the characters into code that can be used for data processing, OCR is sometimes also referred to as text recognition.
OCR systems are made up of a combination of hardware and software that is used to convert physical documents into machine-readable text. Hardware, such as an optical scanner or specialized circuit board is used to copy or read text while software typically handles the advanced processing. Software can also take advantage of artificial intelligence (AI) to implement more advanced methods of intelligent character recognition (ICR), like identifying languages or styles of handwriting.
The process of OCR is most commonly used to turn hard-copy legal or historic documents into PDFs. Once placed in this soft copy, users can edit, format and search the document as if it was created with a word processor.

3.4.1 How optical character recognition works

The first step of OCR is using a scanner to process the physical form of a document. Once all pages are copied, OCR software converts the document into a two-color, or black and white, version. The scanned-in image or bitmap is analyzed for light and dark

areas, where the dark areas are identified as characters that need to be recognized and light areas are identified as background.
The dark areas are then processed further to find alphabetic letters or numeric digits. OCR programs can vary in their techniques, but typically involve targeting one character, word or block of text at a time. Characters are then identified using one of two algorithms:

1. Pattern recognition- OCR programs are fed examples of text in various fonts and formats which are then used to compare, and recognize, characters in the scanned document.

2. Feature detection- OCR programs apply rules regarding the features of a specific letter or number to recognize characters in the scanned document. Features could include the number of angled lines, crossed lines or curves in a character for comparison. For example, the capital letter “A” may be stored as two diagonal lines that meet with a horizontal line across the middle.

When a character is identified, it is converted into an ASCII code that can be used by computer systems to handle further manipulations. Users should correct basic errors, proofread and make sure complex layouts were handled properly before saving the document for future use.

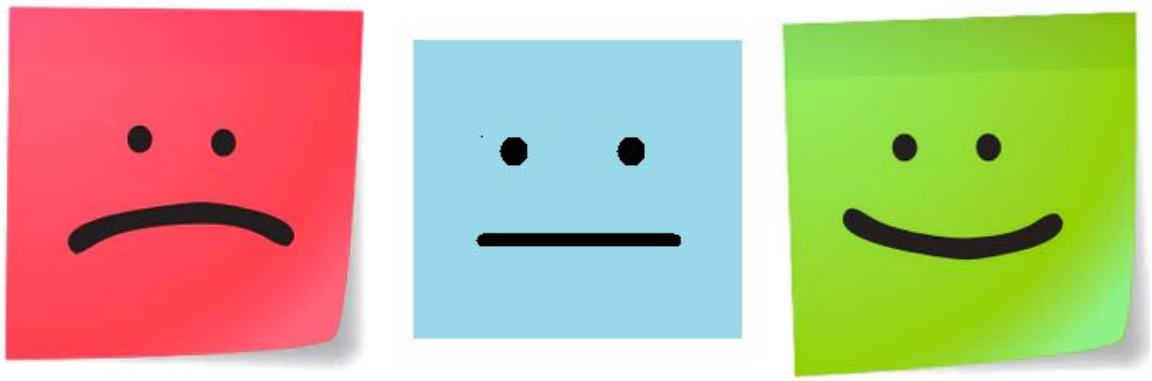
3.4.2 Why we use it at Athena Assistant?

We use OCR to read any soft-copy for the User in Arabic or English.

3.4.3 The implementation:

We will perform both (1) text detection and (2) text recognition using OpenCV, Python, and Tesseract.
We perform text detection using OpenCV’s EAST deep learning model. Using this model, we were able to detect and localize the bounding box coordinates of text contained in an image.
The next step is to take each of these areas containing text and actually recognize and OCR the text using OpenCV and Tesseract.
Because of the long request time we will use for the English OCR the google offline API and for the Arabic OCR we will use the end-point which we build using OpenCV and Tesseract, but this take at least 8 sec and the time to upload the image.

3.5 Sentiment Analysis



Text sentiment refers not to the cut and dry meaning of text, but rather the feeling, attitude, and opinion behind it: “Is this movie review positive, negative, or neutral? Is this customer praising or criticising their purchase?” Extracting sentiment from a body of text to determine the writer’s attitude is commonly known as sentiment analysis or opinion mining.

Although sentiment analysis appears in various places, it is particularly useful in social media. To users, social media is a platform for thought-sharing, which in turn breeds discussion within a community. To businesses, social media is a platform for user feedback. The sheer amount of user feedback available allows businesses to perform data analysis, like sentiment analysis, to measure overall customer satisfaction (and dissatisfaction towards one’s product and services, quantify how effective one’s marketing campaigns are, and help adjust/drive future product and marketing goals. Ergo, sentiment analysis on social media not only benefits customers and businesses, but also shifts business models in the long run. To read more about how sentiment analysis can play a role in business.

Nonetheless, there are some limitations to sentiment analysis. Because language is often expressed with tone, sarcasm, and irony, or with images, videos, and emojis in juxtaposition and contradiction, sentiment analysis is not perfect. However, it remains very useful and will only improve as machine learning techniques advance.

3.6 CHATBOT

A **chatbot** is an artificial intelligence-powered piece of software in a device (Siri, Alexa, Google Assistant etc), application, website or other networks that try to gauge consumer's needs and then assist them to perform a particular task like a commercial transaction, hotel booking, form submission etc .Today almost every company has a chatbot deployed to engage with the users. Some of the ways in which companies are using chatbots are:

- To deliver flight information
- to connect customers and their finances
- As customer support

The possibilities are (almost) limitless.

History of chatbots dates back to 1966 when a computer program called ELIZA was invented by Weizenbaum. It

imitated the language of a psychotherapist from only 200 lines of code. You can still converse with Eliza.

There are broadly two variants of chatbots:

Rule-Based and **Self learning.**

1. In a **Rule-based approach**, a bot answers questions based on some rules on which it is trained on. The rules defined can be very simple to very complex. The bots can handle simple queries but fail to manage complex ones.
2. The **Self learning bots** are the ones that use some Machine Learning-based approaches and are definitely more efficient than rule-based bots. These bots can be of further two types: **Retrieval Based** or **Generative**

i) In **retrieval-based models**, a chatbot uses some heuristic to select a response from a library of predefined responses. The chatbot uses the message and context of conversation for selecting the best response from a

predefined list of bot messages. The context can include a current position in the dialog tree, all previous messages in the conversation, previously saved variables (e.g. username). Heuristics for selecting a response can be engineered in many different ways, from rule-based if-else conditional logic to machine learning classifiers.

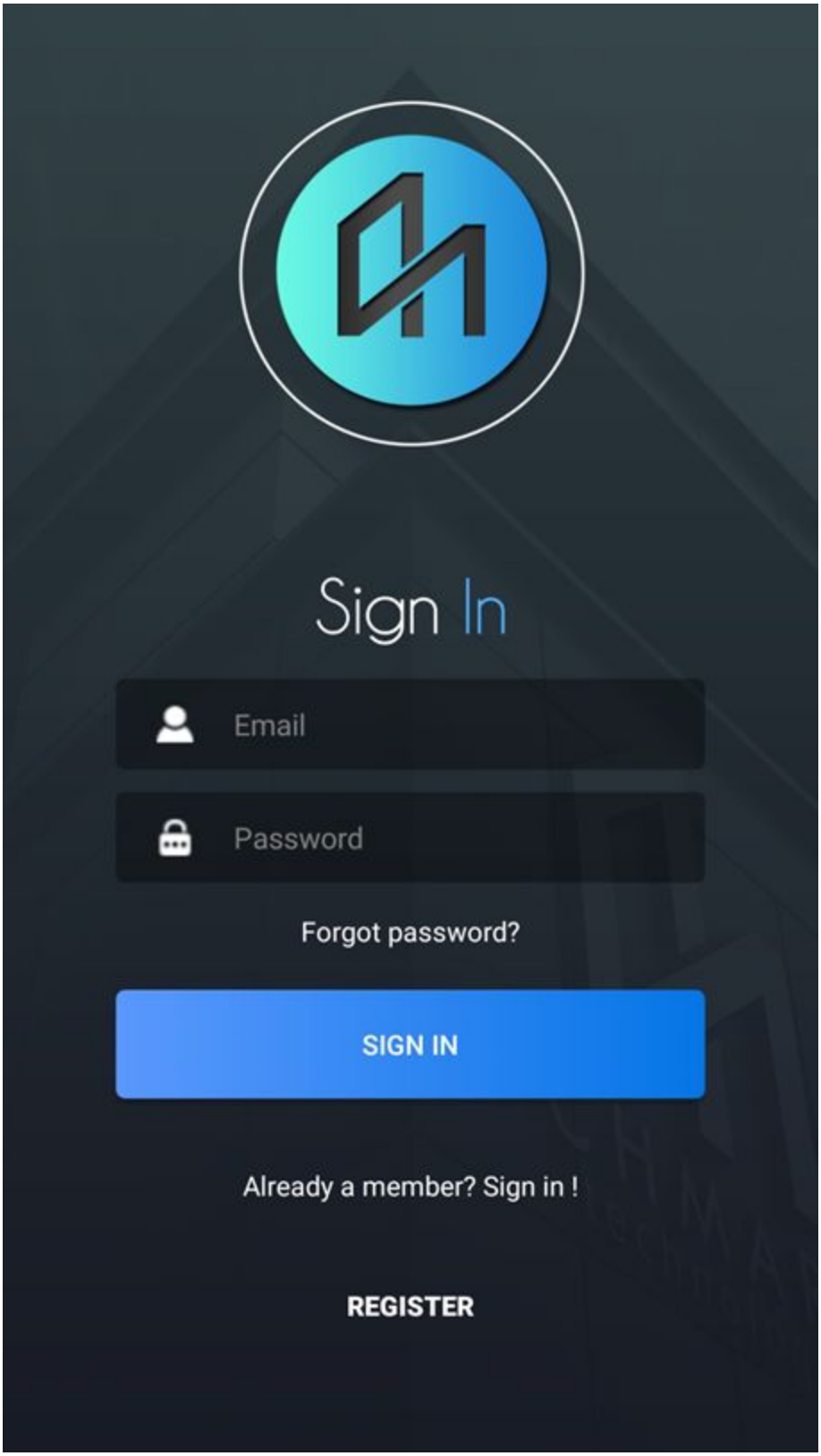
ii) **Generative** bots can generate the answers and not always replies with one of the answers from a set of answers. This makes them more intelligent as they take word by word from the query and generates the answers.

Chapter Four

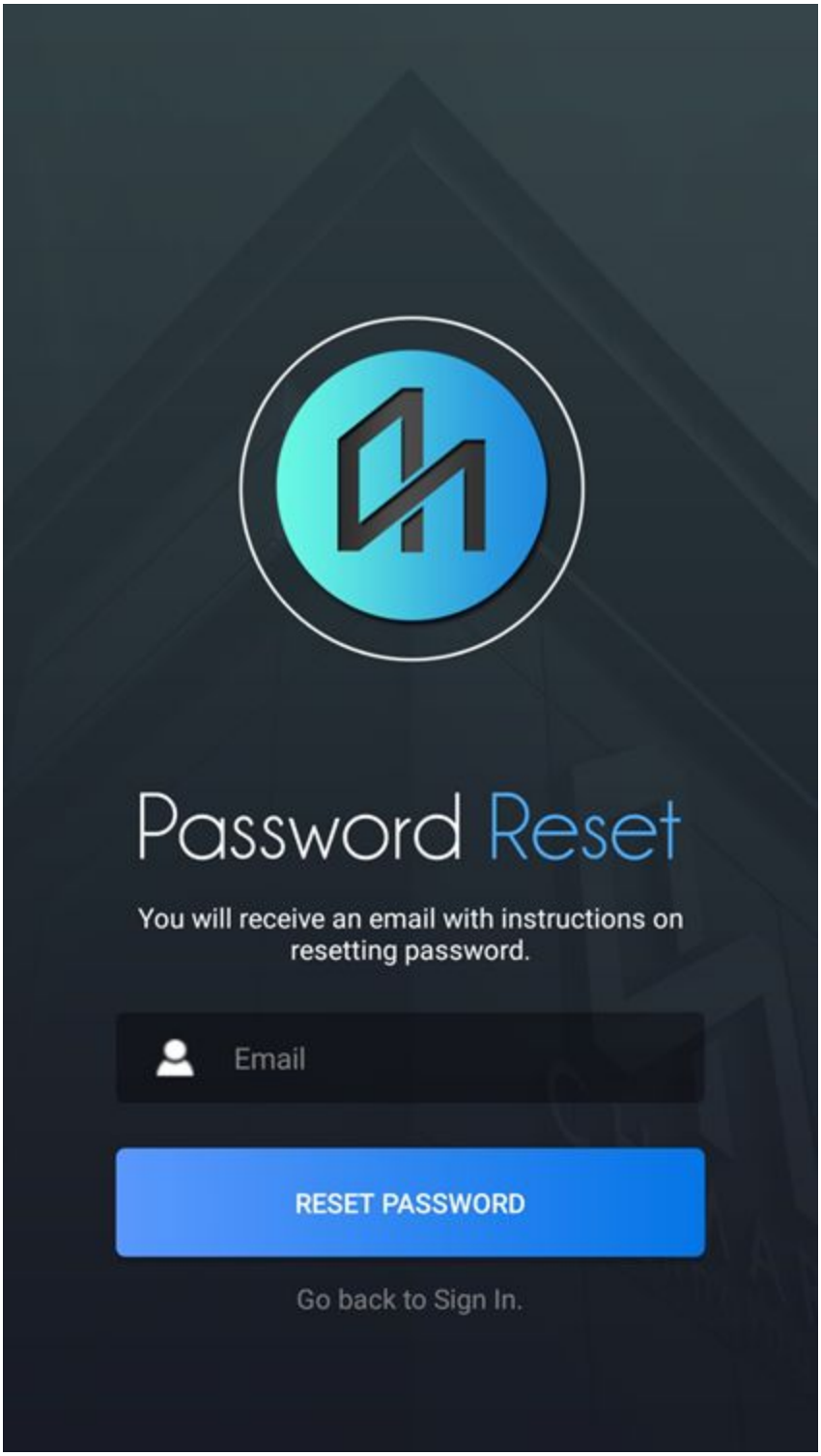
4 Implementation Results

Splash Screen of Our Application

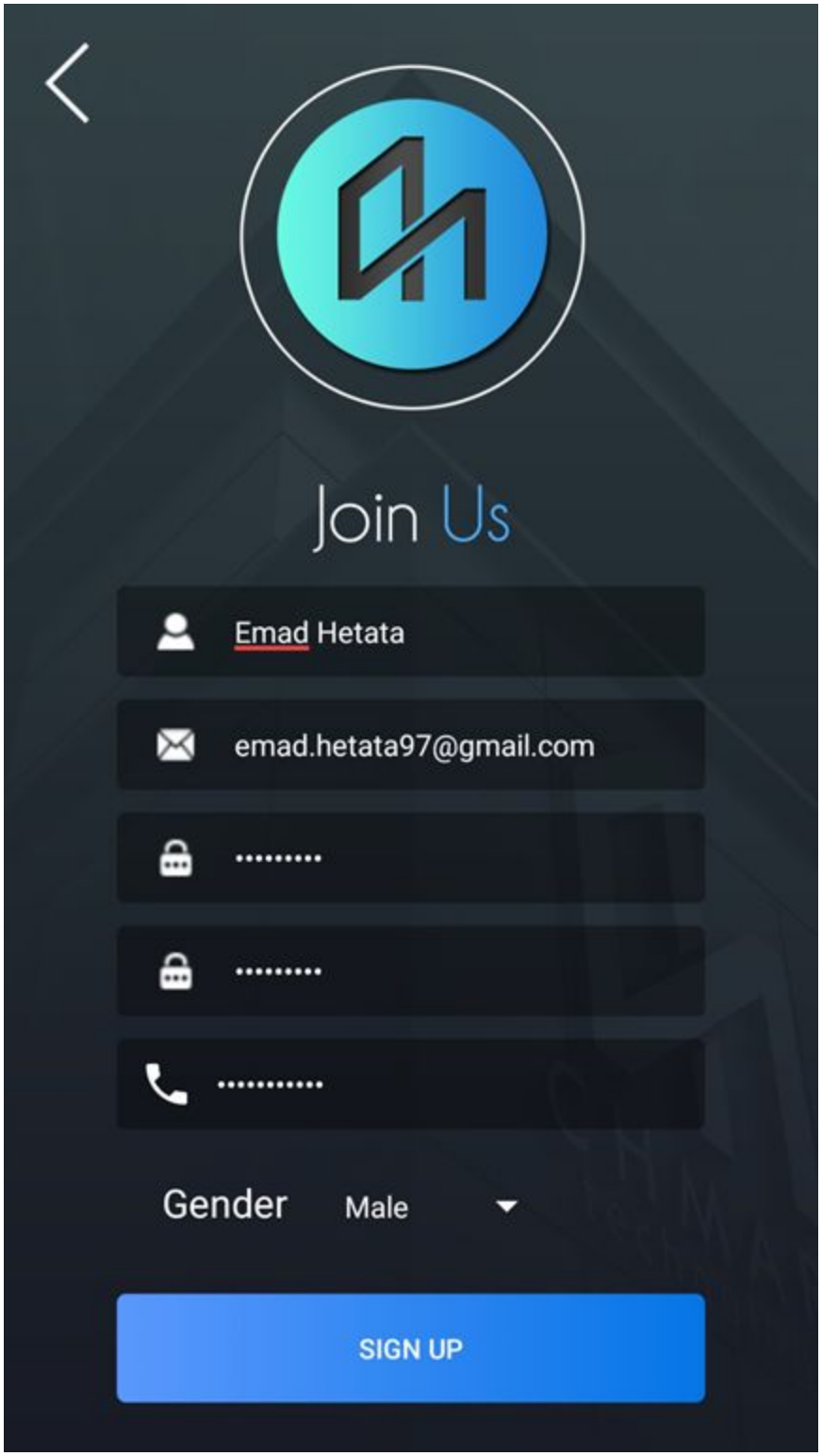
One of 3 Pages for introduce application



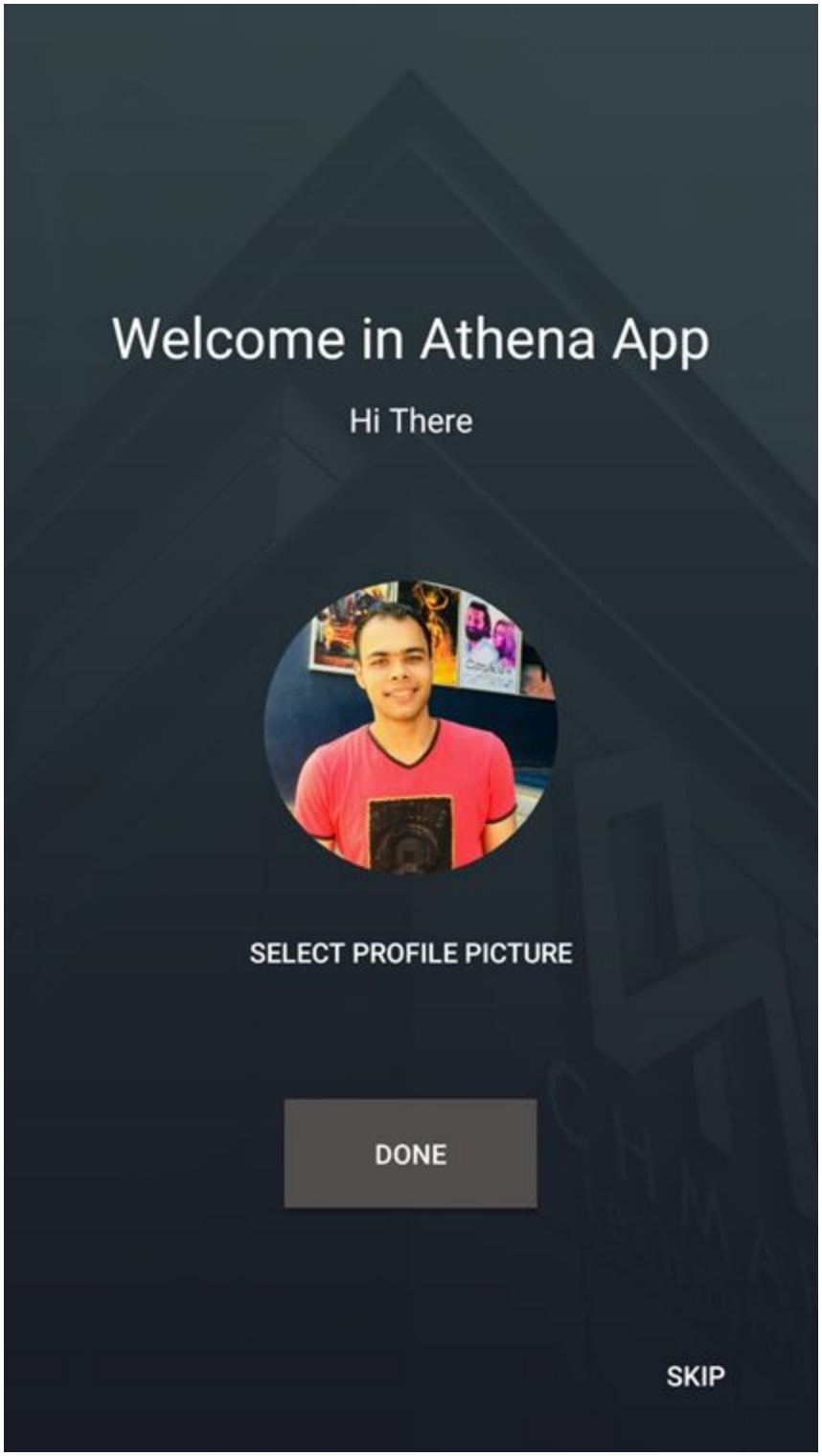
Here User can Login or navigate to Register



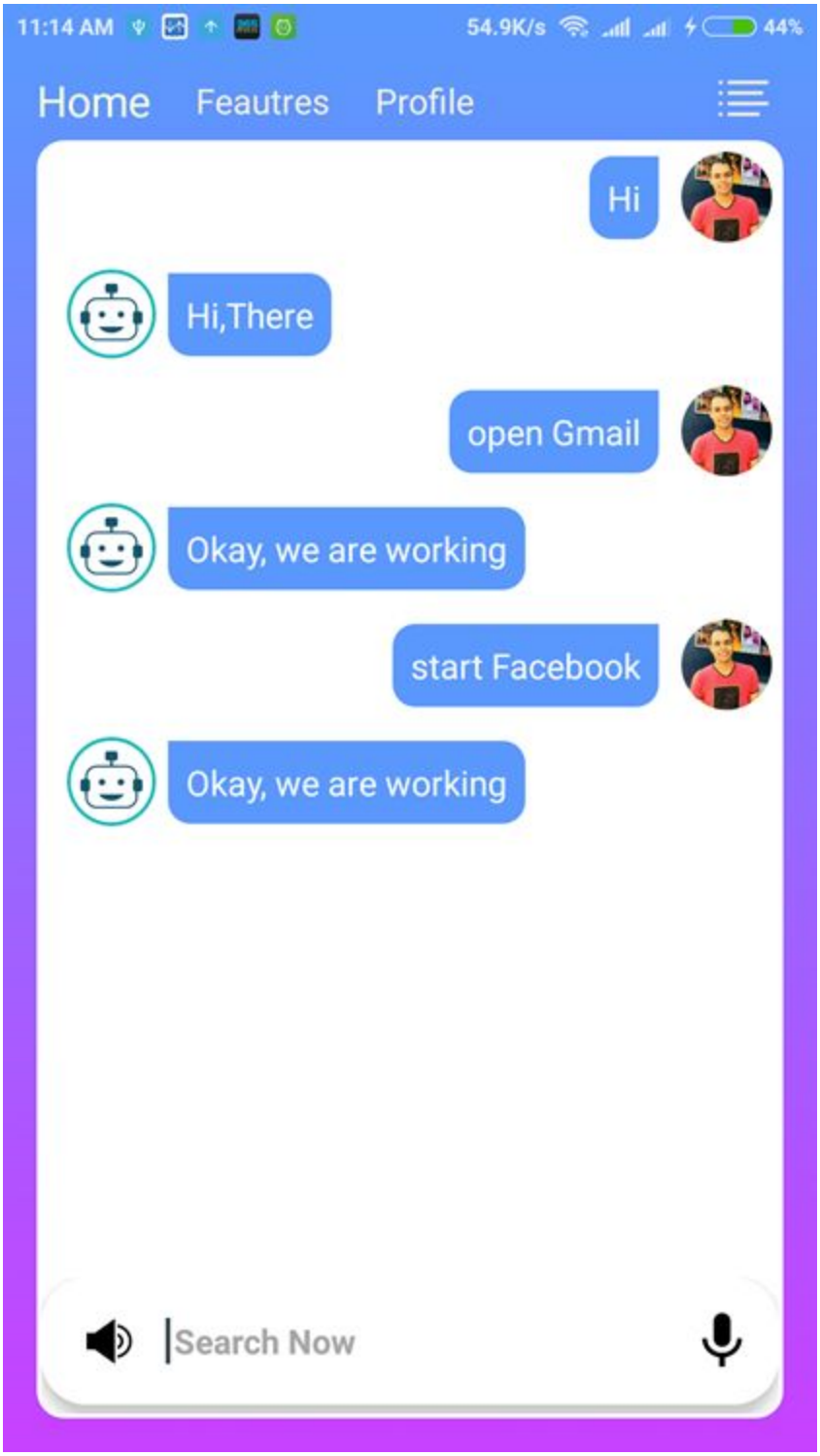
Here User can reset his Password



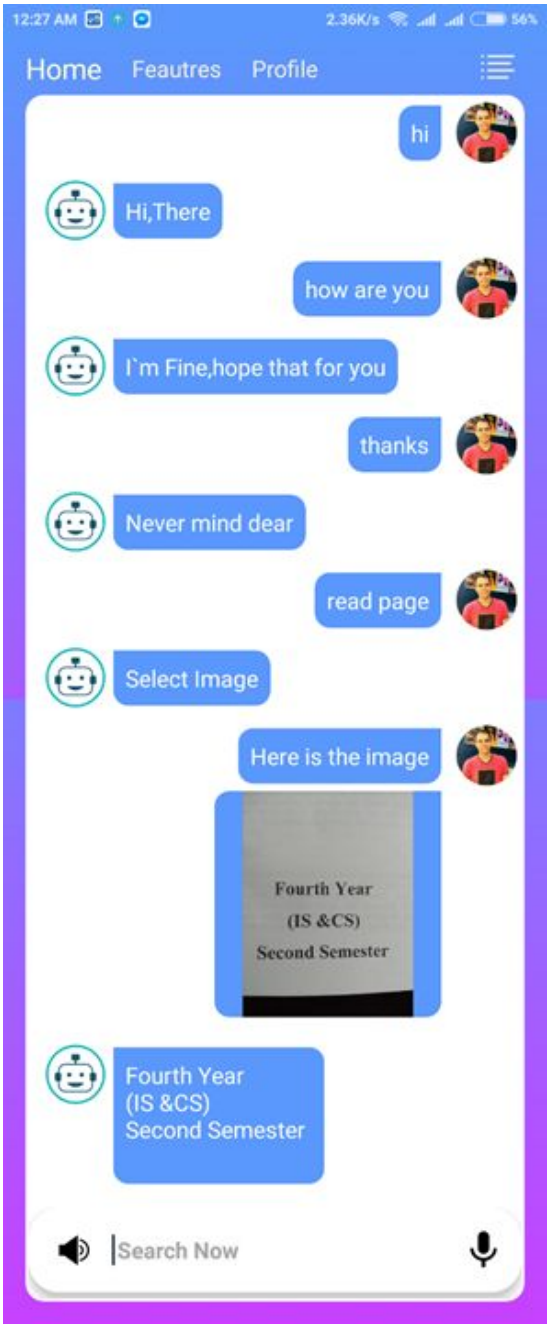
Here User Can Register New Account



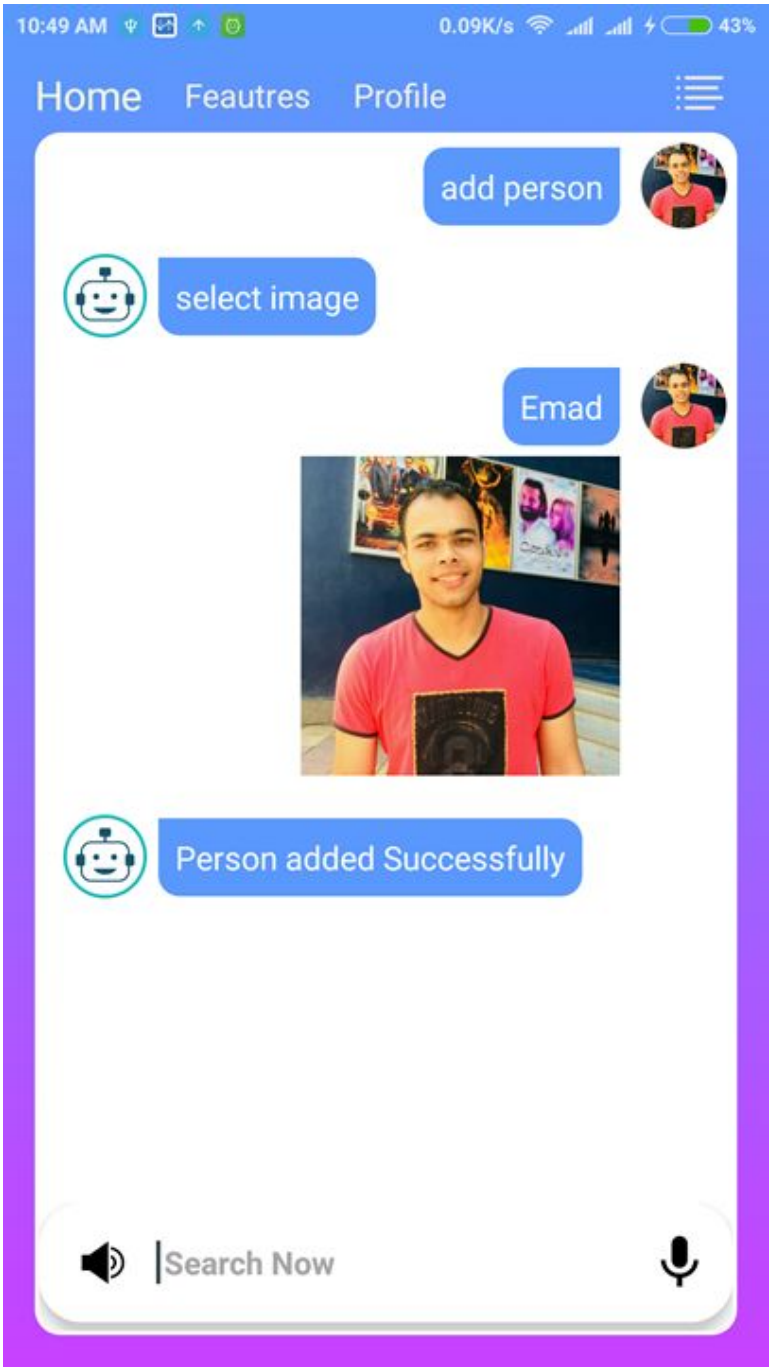
After Creating new Account, you can Select your Profile Image.



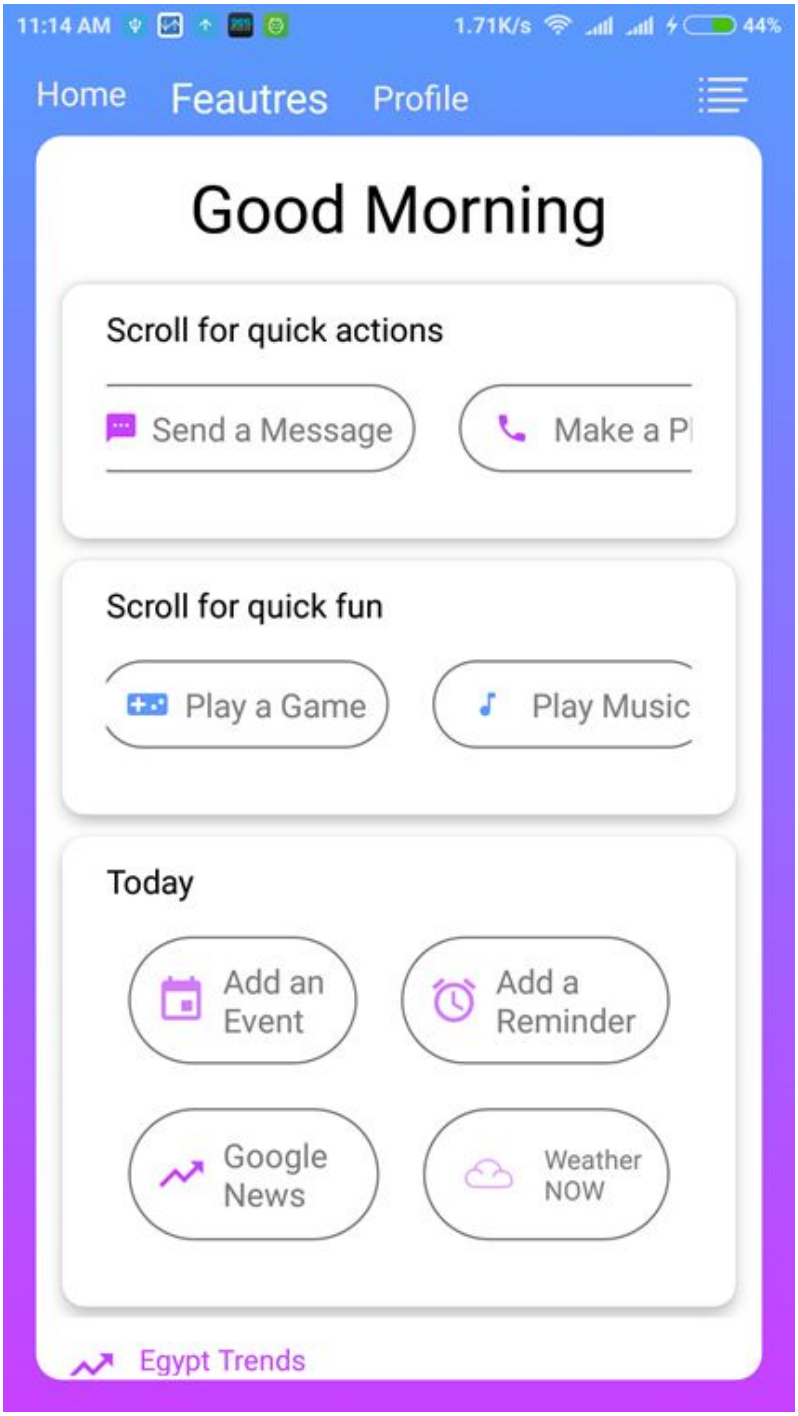
Simple Queries like opening any app installed on your phone.



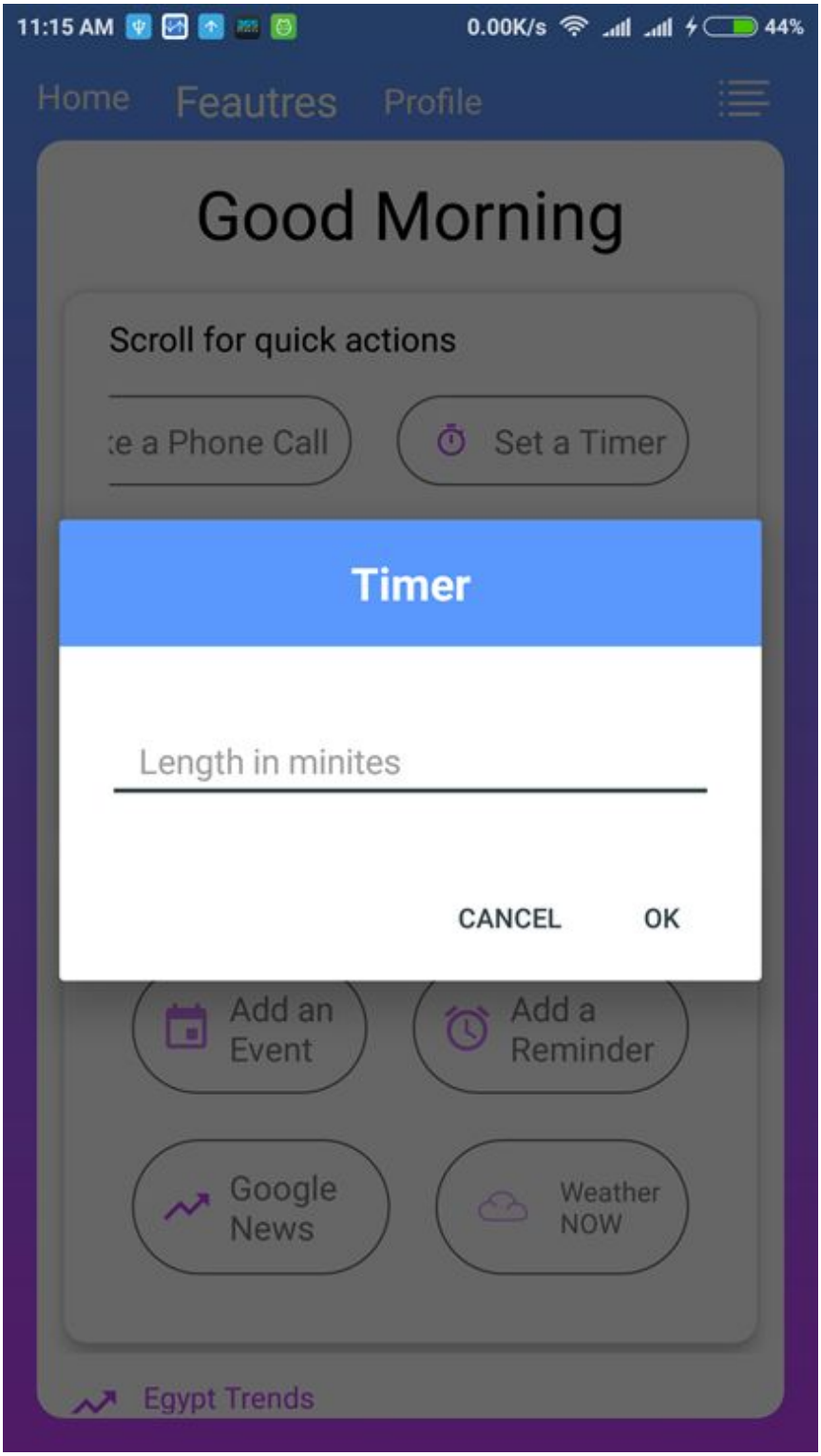
One of our main Functions is Optical Character Recognition in Arabic and English(Computer Written).



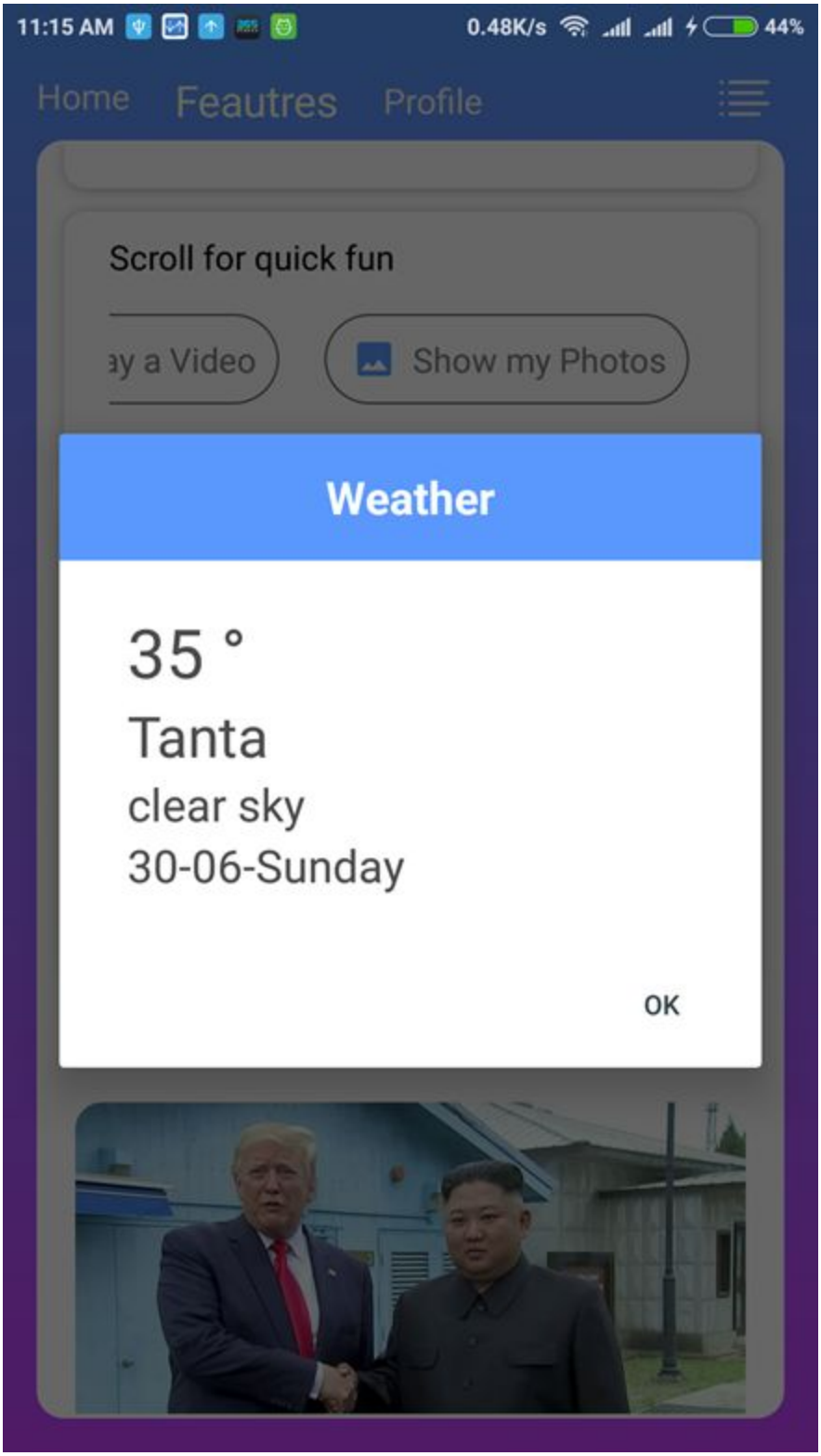
Another Functionality is training our model to allow it to recognize Person (Face Recognition).



Features Page Provides user some quick actions and User Country Trends.



Ex: Set new Timer

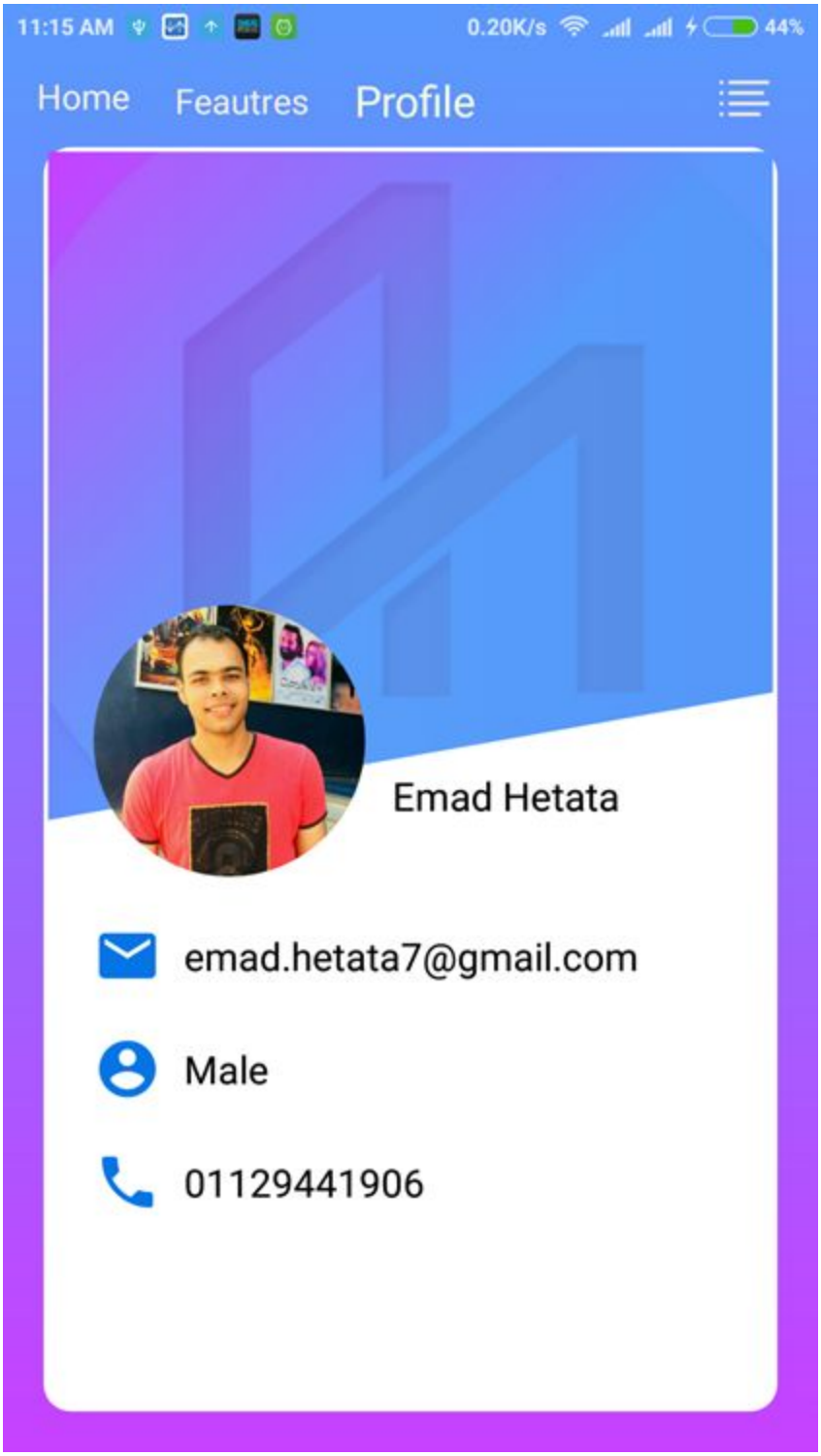


Ex: Weather of User City

Ex: Provide Trends

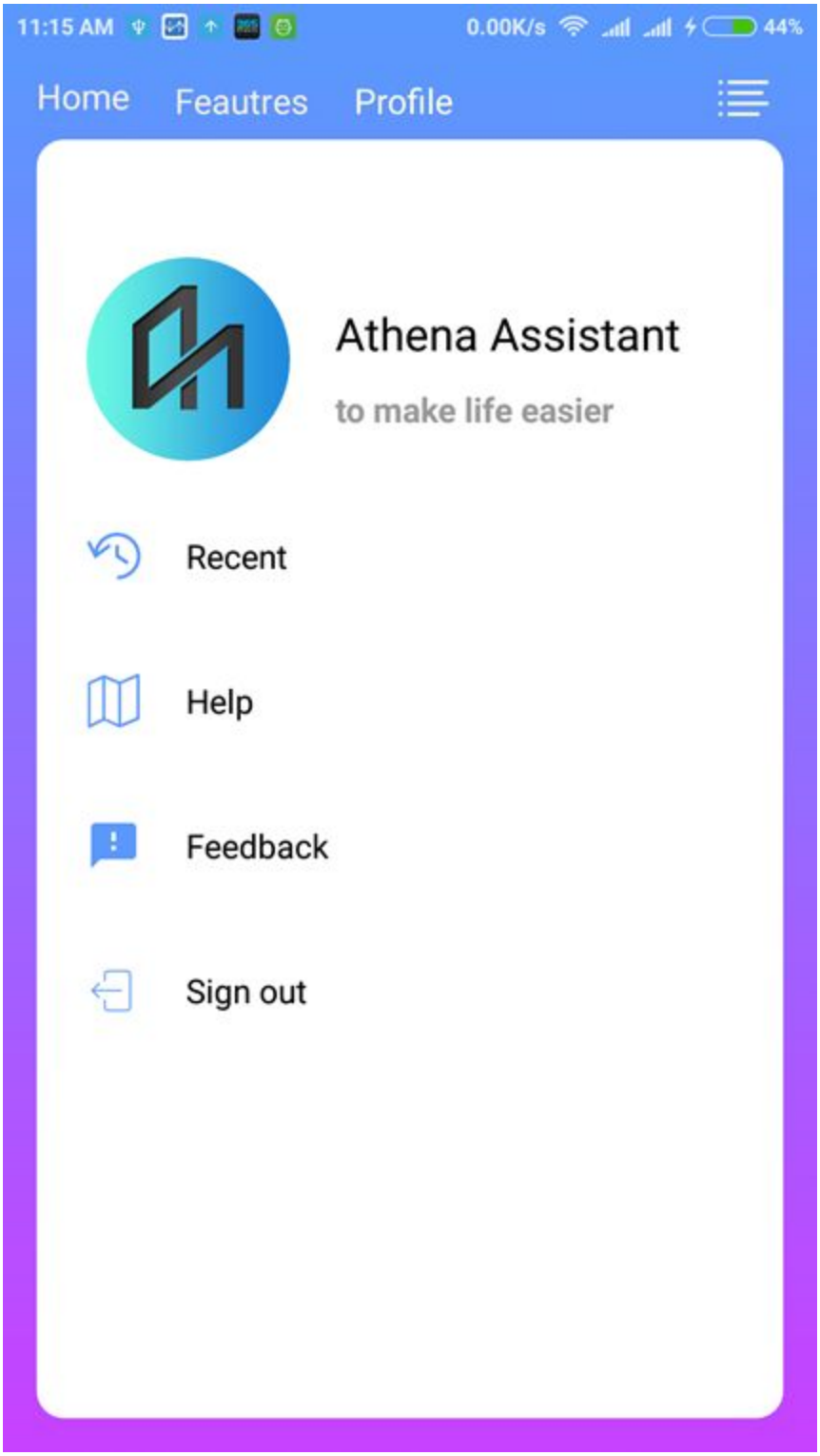
User can interact with the news

And See the trend if he needs to read full trend click see more.

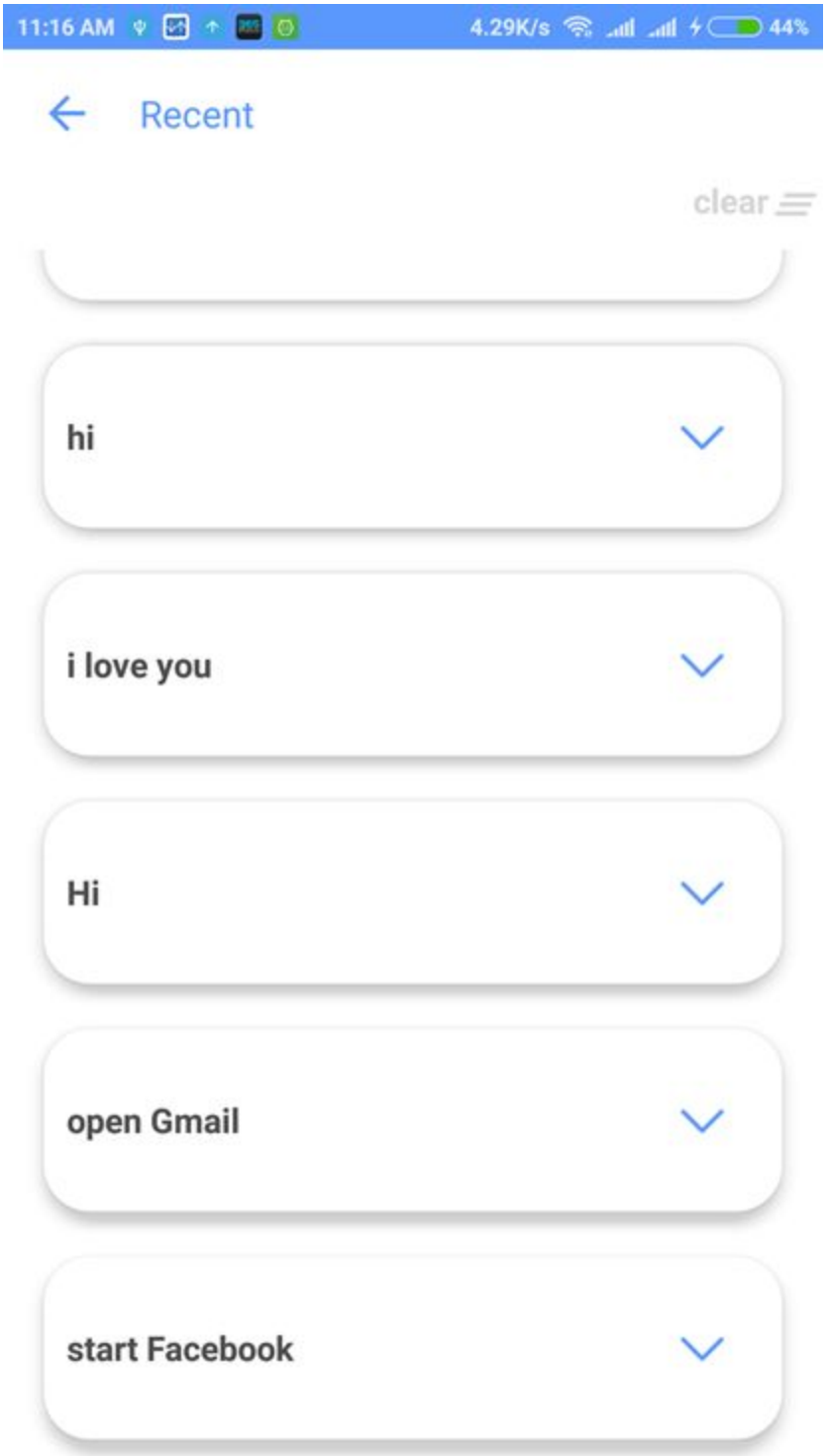


Profile Page

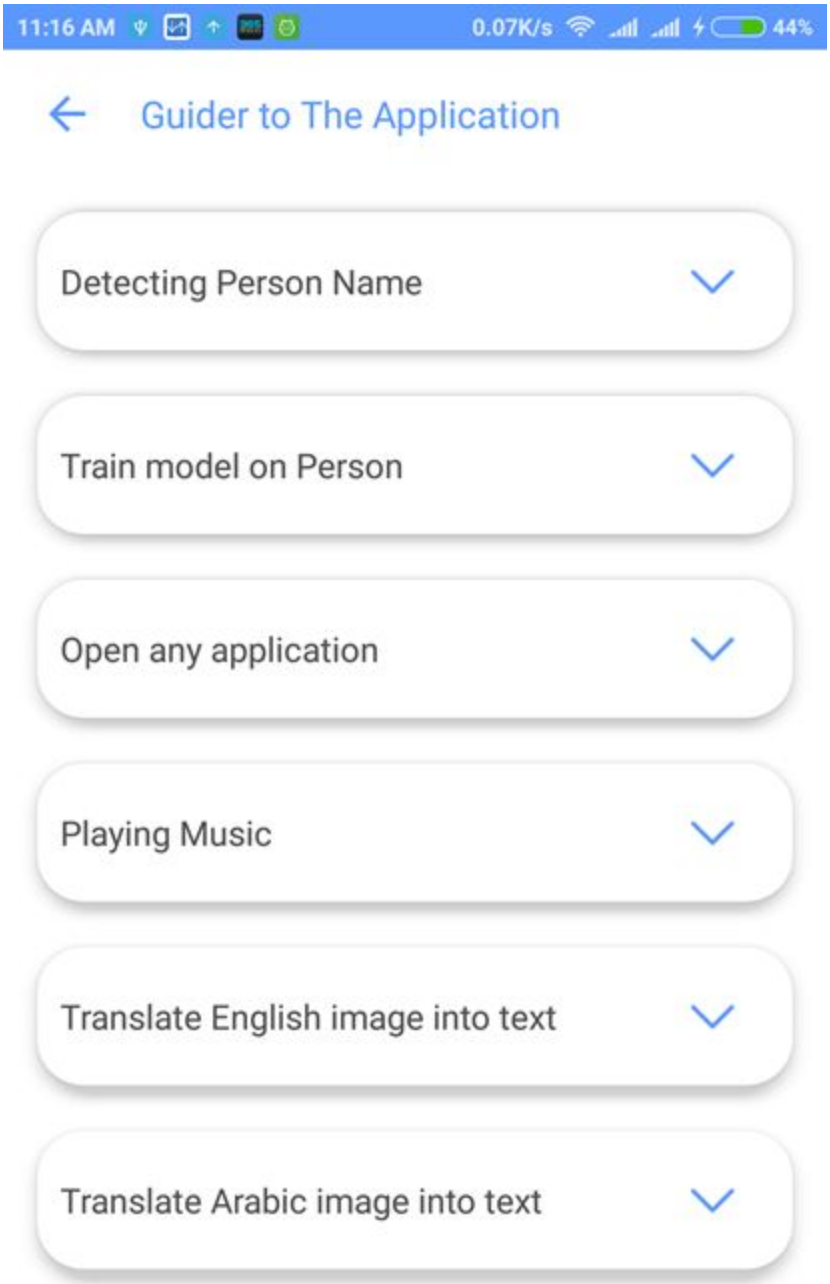
Provides user information.



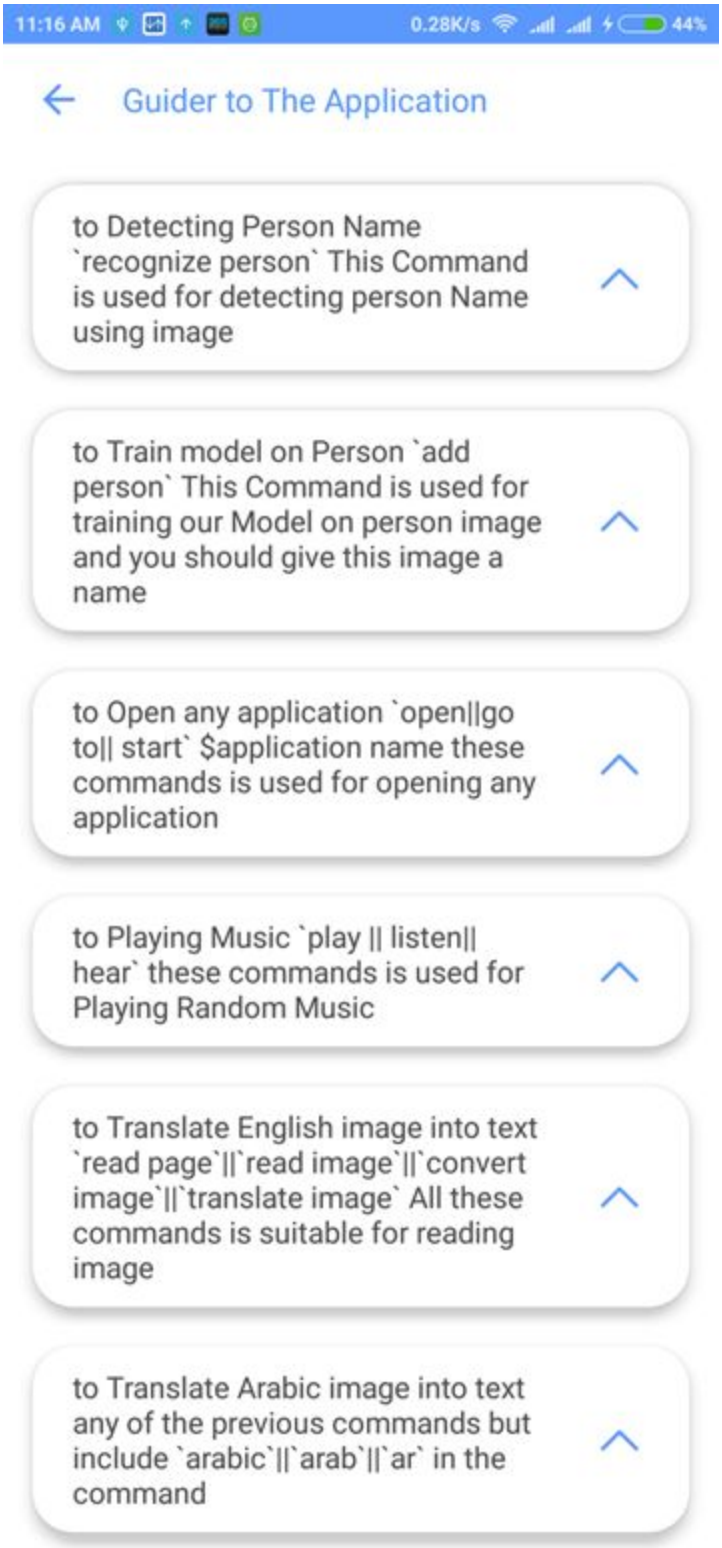
More Tools like Recent Queries and more.



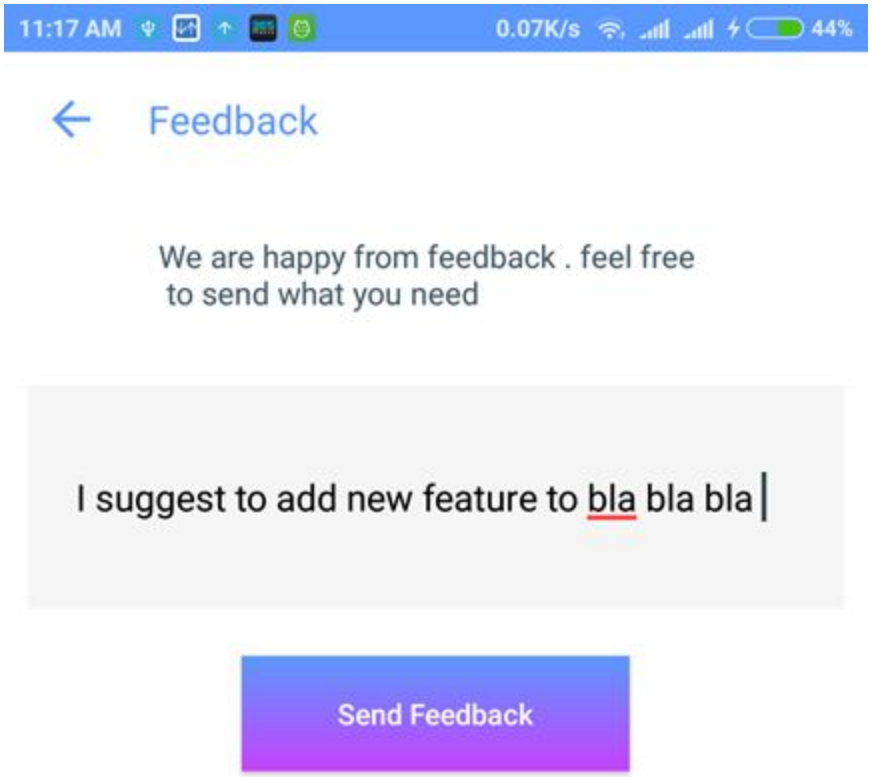
Recent Queries by the current user, also user can delete or copy any query.



Guide to Application in Collapsed mode.



Guide to Application in Expanded mode.



User can give Feedback to us.

Chapter Five

5 CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

As We went deep through machine learning and deep learning techniques to develop a personal assistant that depends on some basic skills such as Speech Recognition, Natural Language Processing , Sentiment Analysis and Image Classification. We found out that it's all about how much data your assistant has trained on, and how these data are represented so the more data you train your assistant, the more accuracy you will get and the more intelligent assistant you will make.

5.2 FUTURE WORK

As mentioned before, our main problem was to collect enough data to train our assistant to be more reliable and intelligent and use appropriate algorithms to be trained on data.
So, in the future we will focus on collecting more datasets and training our assistant trying to reach higher accuracy and reliability to make a project that could compete in the global market.

Chapter Six

6 REFERENCES

6.1 OCR References

- https://developers.google.com/android/reference/com/google/android/gms/vision/text/TextRecognizer
- https://www.pyimagesearch.com/opencv-tutorials-resources-guides/
- https://github.com/tesseract-ocr

6.2 Speech Recognition References

- https://realpython.com/python-speech-recognition/#putting-it-all-together-a-guess-the-word-game
- https://en.wikipedia.org/wiki/Speech_recognition#Deep_feedforward_and_recurrent_neural_networks
- https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a
- https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6?source=bookmarks---------1----------
- https://www.amazon.com/Machine-Learning-Python-Cookbook-Preprocessing/dp/1491989386

6.3 Face Recognition References

- https://www.bytefish.de/pdf/facerec_python.pdf
- https://www.rankred.com/face-recognition-algorithms-techniques/
- http://eyalarubas.com/face-detection-and-recognition.html
- https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html
- https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors#Principal_components_analysis
- https://en.wikipedia.org/wiki/Local_binary_patterns

- https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b
- https://docs.opencv.org/3.4.3/d7/d8b/tutorial_py_face_detection.html
- https://docs.opencv.org/2.4/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html
- https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#face-detection
- https://thecodacus.com/face-recognition-opencv-train-recognizer/

6.4 Android References

- https://firebase.google.com/docs/android/setup
- https://www.udemy.com/topic/android-development

6.5 Chatbot References

- https://medium.com/analytics-vidhya/building-a-simple-chatbot-in-python-using-nltk-7c8c8215ac6e
- https://en.wikipedia.org/wiki/Similarity_measure
- https://triton.ml/blog/sentiment-analysis
- https://towardsdatascience.com/creating-the-twitter-sentiment-analysis-program-in-python-with-naive-bayes-classification-672e5589a7ed
- https://medium.com/datadriveninvestor/implementing-naive-bayes-for-sentiment-analysis-in-python-951fa8dcd928
- https://medium.com/datadriveninvestor/implementing-naive-bayes-for-sentiment-analysis-in-python-951fa8dcd928