Call Center Automation Using Speaker Recognition

By

Abhi Shah 14BCE107
Rachit Shah 14BCE110

Abhi Shah 14BCE107
Rachit Shah 14BCE110

DEPARTMENT OF COMPUTER ENGINEERING
Ahmedabad 382481

Call Center Automation Using Speaker Recognition

Mini Project - 3

Submitted in fulfillment of the requirements

For the degree of

Bachelor of Technology in Computer Engineering/Information Technology

By

Abhi Shah 14BCE107
Rachit Shah 14BCE110

Guided By
Prof. Monika Shah
DEPARTMENT OF COMPUTER ENGINEERING
Ahmedabad 382481

CERTIFICATE

This is to certify that the Seminar entitled "Call Center Automation Using Speaker Recognition" submitted by Abhi Shah(14BCE107),Rachit Shah(14BCE110) towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Engineering of Nirma University is the record of work carried out by him/her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Prof. Monika Shah   Dr, Sanjay Garg
Assistant Professor   Head Of Dept. of Computer Engg.,
Department of Computer Engg.,   Institute of Technology,
Institute of Technology,   Nirma University,
Nirma University,   Ahmedabad
Ahmedabad

ACKNOWLEDGEMENT

First of all, we would like to pay our greetings to our guide, Prof. Monika Shah who helped and supported us in completing this course. She guided us throughout the semester. We would also like to pay our gratitude towards the institute for keeping such a course in which the students are allowed to explore a technology or a field of their own choice as a part of their curriculum. We would also like to acknowledge our parents for supporting us throughout our life.

ABSTRACT

Our Project involves call center automation web application to avoid various difficulties which arises because of unavailability of the agents. We used Python Language and Convolutional Neural network algorithms to complete our task. Our web application aims to create agents and allot them tasks using queue mechanism.We used Twilio API for that purpose.We also used Convolutional Neural Network Algorithm to recognize customers without taking their details every time.So Customer data can be available to agents automatically using speaker recognition.

Table of Contents

Table of Figures

Chapter 1: Introduction

1.1 General

A Call Center is a centralized approach where a number of workers are attendants whose duty is to attend to the complaints of customers whether via phone, web chat or video call. These complaints differ according to the organization for whom the call center is working for. For example, organizations like Internet Service Providers (ISPs) may use call centers to solve customers problems like

- Billing information
- Cable Cut issue
- Refund issues
- Cancellation of service
- Network down
- Router/modem issues
- Offer inquiry
- Miscellaneous queries

While a banking call center might specialize in queries/complains regarding:

- Transaction status
- New card application
- Cheque withdrawals
- International card application
- Internet Banking
- Fund Transfers
- Others

With each domain, the specialization of call centers changes. Our project focuses on the call center for ISPs. It focuses on minimizing the workload for human agents by automating questions for the customer such that easy problems are solved without any human intervention. After that if the customer's problem still hasn't been solved then the call can be arranged to a worker/attendant at the call center.

Our project also focuses on the machine learning aspect by predicting the speaker who is speaking using Convoluted Neural Networks (CNNs) and grabbing its data from the database like past problems to decrease the work for human agent and also filter out notorious fraud callers who might be calling by impersonating people.

1.2 Objective of Study

Our main objective to choose this application for our mini project was mainly because we were interested in artificial neural network as well as database design and thought it would help us in our future endeavours.We wanted to learn neural networks using python. We were always impressed by recognition systems made using neural network, how this applications work internally and we always wanted to create these applications on our own and study various python concepts.We were also interested in automation techniques that can be used in call center to reduce human intervention.This mini project gave us the opportunity to pursue our goals and objectives.So in this mini project we have started to learn various python concepts and APIs.

1.3 Scope of Work

This project covers various functionalities needed by call centers. Nowadays, we have to wait for very long time to connect with the agents. Using this automation system, most of the common problems can be solved automatically.So It will allow customers to solve their problems easily. It will also reduce the workload of human agents and thus maximize the utility of the call center.

Chapter 2: Literature Survey

2.1 General

As mentioned above we mainly focus on speaker recognition and automation using convolutional neural network in Python. And automating customer requests to solve easy problems and implement chat, phone and video functionality in the call center.

2.2 Literature Review

There are a number of tutorials available for using Tensorflow in our project. Tensorflow.org [4] presents an easy guide right from installing it to using it in various applications. It provides a number of inbuilt libraries and methods to make our work easier and also efficient.

Keras [5] is a library for python which uses tenorflow and theano at backend for deep learning applications. It provides a user-friendly, modular and extensible way to implement CNN and RNN networks. Its guide is also quite extensive.

"Urban Sound Classification" by Aqib Saeed [6] shows implementing neural networks, CNNs and RNNs to classify sounds like dogs, cats, cars, etc. It uses a handmade dataset of sounds of dogs, cats, cars, etc. and the using it to predict new sounds. It extracts features using the MFCC algorithm and then feeds it to a neural network with 2 hidden layers. It achieved a F-score of 82%.

Twilio is a cloud service which provides for APIs to implement voice calls, chats and video calls in any application. Its tutorials [7] provide an extensive guide about how to create workspaces, how to assign workers and their tasks, task queues, worker queues, etc.

Heroku is a cloud service which helps to deploy apps much like Amazon AWS, Google Cloud,etc. The Heroku DevCenter [8] provides an easy guide to deploy our applications as a website on the cloud.

Chapter 3: Speaker Recognition

3.1: Languages and Modules Used

- Python 3.5.3
- Jupyter Notebook
- Tensorflow
- Keras
- Sklearn
- Librosa (for feature extraction using MFCC)
- H5py
- Pickle

3.2: Implementation

Artificial Neural Network Model : We first implemented simple artificial neural network model. We used multilayer perceptron model of two layers with 480 & 500 neurons in each layer. We were able to get 90% accuracy after 60000 epochs.
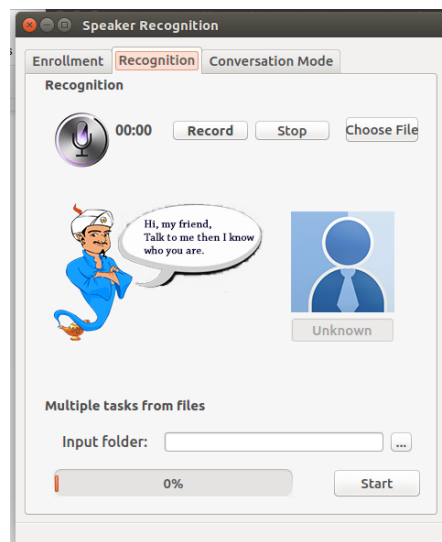

Figure 1 GUI for predicting speaker

In this GUI,We are able to record sound of users & can predict users with accuracy over 92% using convolutional neural network.

Convolutional Neural Network:In convolutional neural network model, first we extracted features of 60*41 in 2 channels. Each audio clip is divided in segments of 60*41. Then these segments are passed to feature extraction module to extract features. We used MFCC (Mel-frequency cepstral coefficients) and logistic coefficients of the sound to identify various features of audio.

We used 193 features from each audio clip. We implemented this CNN model using Keras library in python.After 100 epochs we are able to achieve accuracy over 92%. Once we trained the model we can predict speakers very efficiently. New speakers can not be predicted using this method. So new speakers must be enrolled first and then we have to train the model again to add that speaker. In our dataset we used 38 speaker including us.We took various example voices from speakers i.e fast,solo,synchronized etc. In 38 speaker,we are able to get accuracy over 92%.

Chapter 4: Call Center Automation

4.1: Libraries, APIs and Tools Used

- Twilio API for voice, video and web chat
- Heroku cloud deploy
- AngularJS
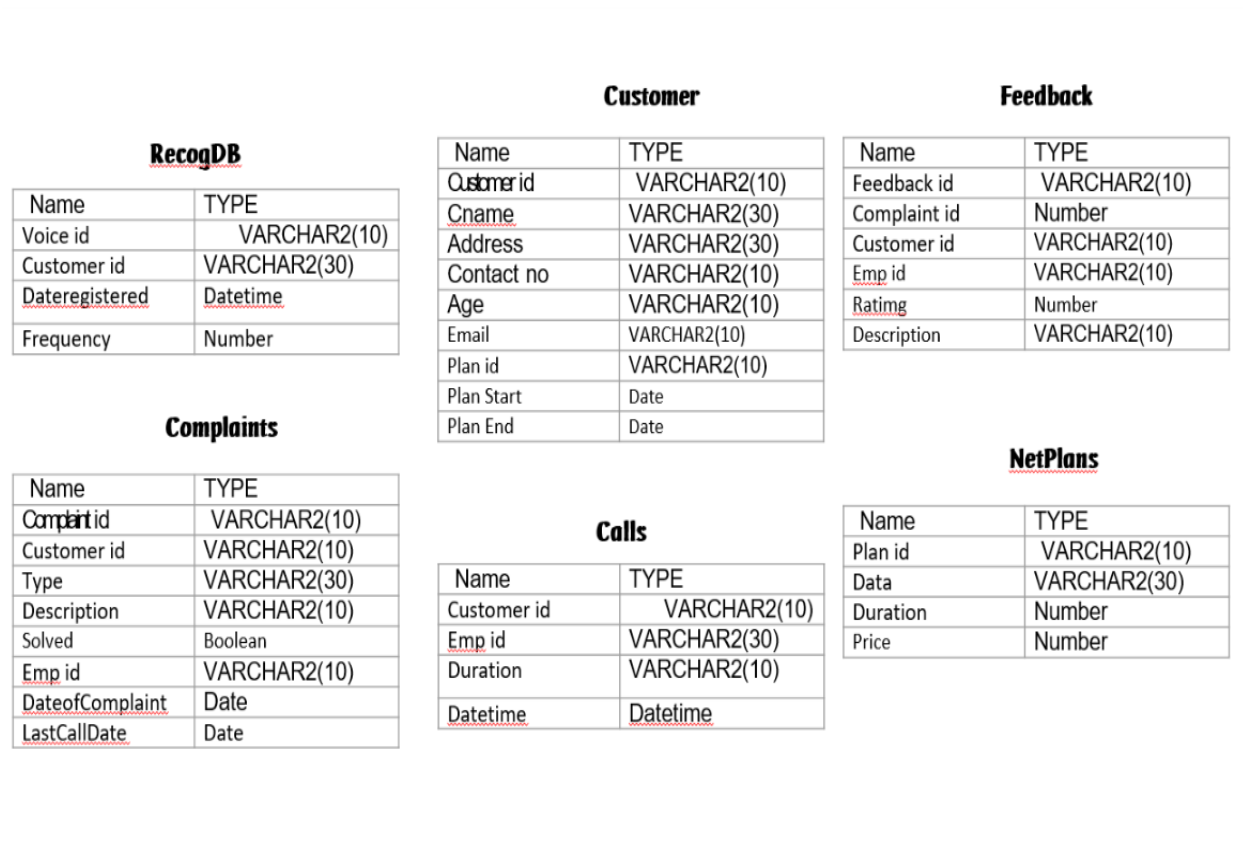
4.2: Database Schema and ER Diagram

4.2.1: Schema:

RecogDB

Name	TYPE
Voice id	VARCHAR2(10)
Customer id	VARCHAR2(30)
Dateregistered	Datetime
Frequency	Number

Complaints

Name	TYPE
Complaint id	VARCHAR2(10)
Customer id	VARCHAR2(10)
Type	VARCHAR2(30)
Description	VARCHAR2(10)
Solved	Boolean
Emp id	VARCHAR2(10)
DateofComplaint	Date
LastCallDate	Date

Customer

Name	TYPE
Customer id	VARCHAR2(10)
Cname	VARCHAR2(30)
Address	VARCHAR2(30)
Contact no	VARCHAR2(10)
Age	VARCHAR2(10)
Email	VARCHAR2(10)
Plan id	VARCHAR2(10)
Plan Start	Date
Plan End	Date

Calls

Name	TYPE
Customer id	VARCHAR2(10)
Emp id	VARCHAR2(30)
Duration	VARCHAR2(10)
Datetime	Datetime

Feedback

Name	TYPE
Feedback id	VARCHAR2(10)
Complaint id	Number
Customer id	VARCHAR2(10)
Emp id	VARCHAR2(10)
Ratimg	Number
Description	VARCHAR2(10)

NetPlans

Name	TYPE
Plan id	VARCHAR2(10)
Data	VARCHAR2(30)
Duration	Number
Price	Number

Figure 2 Database Schema
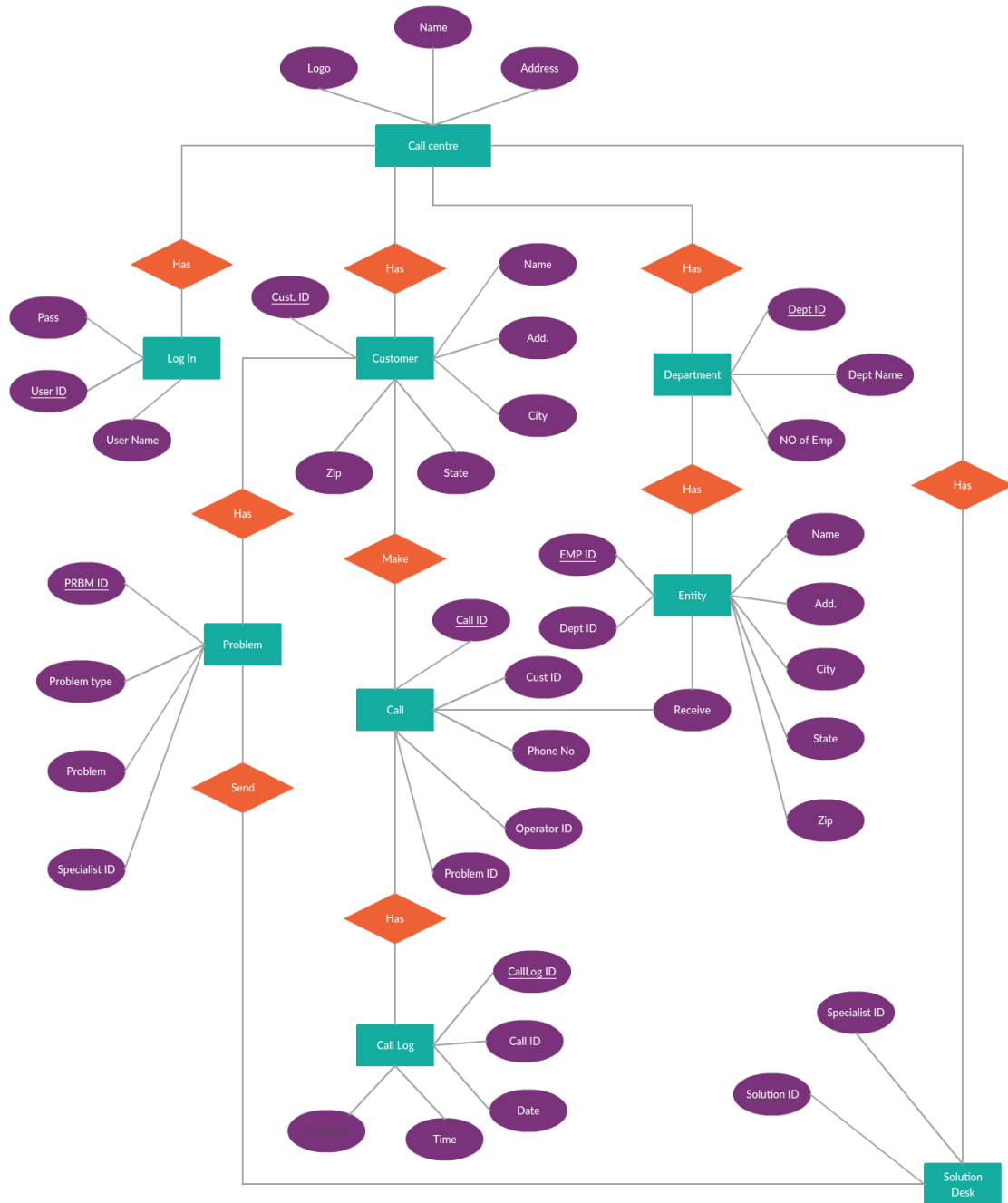
4.2.2: ER Diagram:

Figure 3 ER Diagram

4.3 Use Cases

4.3.1 Use Case 1: Customer has a connectivity problem

Description: We have FAQs listed in our system.Basic Problems can be solved using these FAQs.So that there is no need of any representative intervention.

Solution:FAQs will be asked to customers using text to speech converter.Customer replies to this using yes/no.By using this communication ,problems can be identified for common problems.

4.3.2 Use Case 2: Customer has a problem that is not common

Description: This is in continuity of first use case if problem cannot be solved using first use case.

Solution: Then we can redirect call to call center representative.

4.3.3 Use Case 3: Customer Identification problem

Description: This can be used in parallel for all calls to identify fraudsters,regular customers etc.

Solution:Here we can identify customers using speaker recognition. If the caller is a known prank caller then the representative will be cautioned.

4.4 Flow Charts

4.4.1 Callback Voice Calling
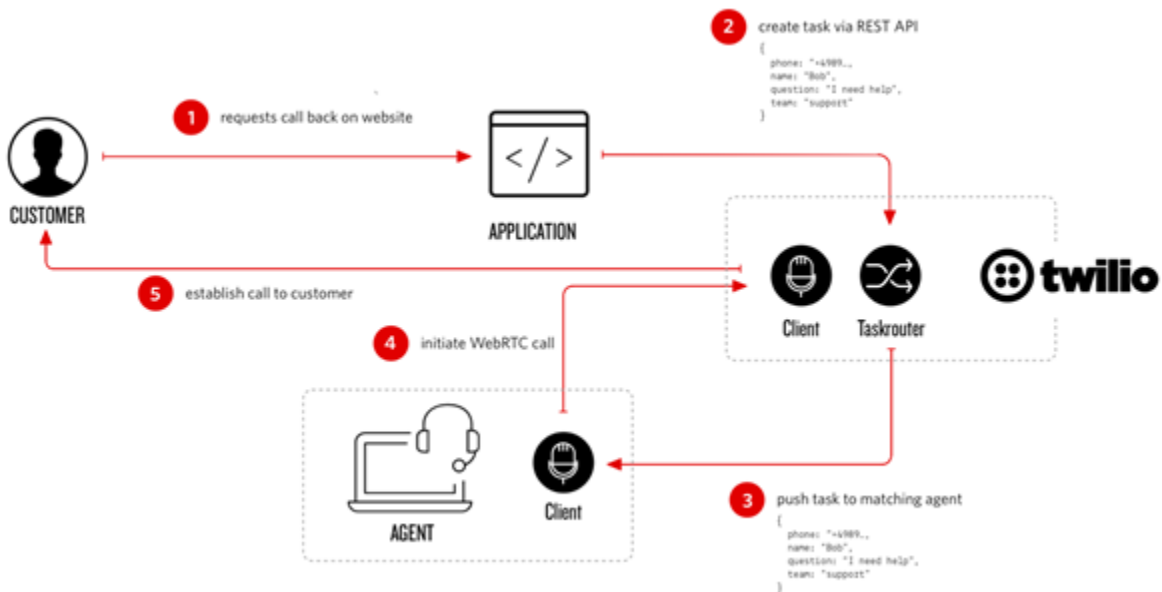
Figure 4 Callback voice calling

4.4.2 Inbound Voice Calling
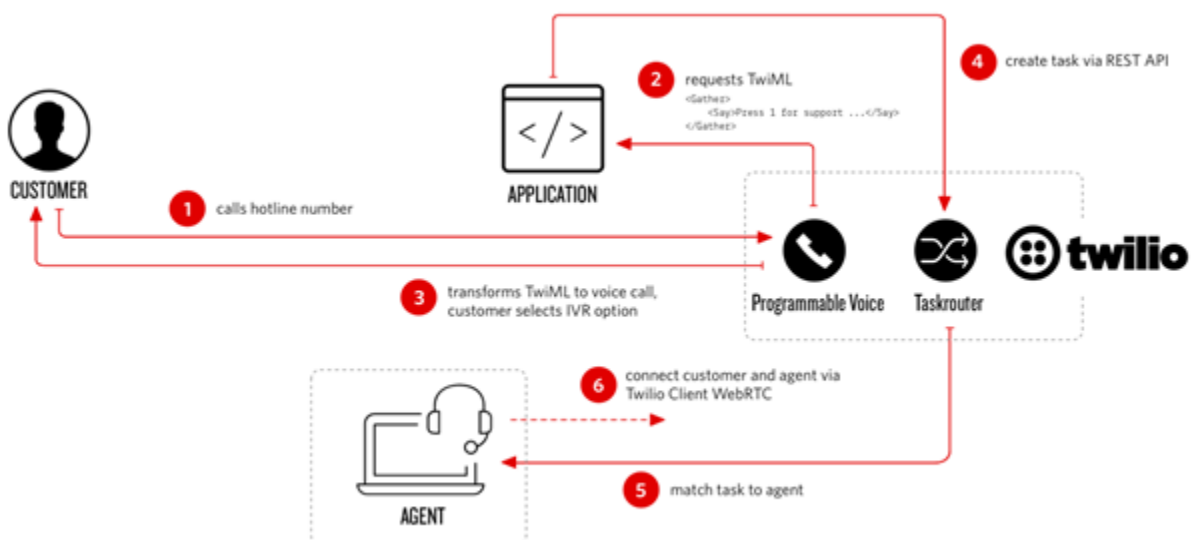
Figure 5 inbound voice calling
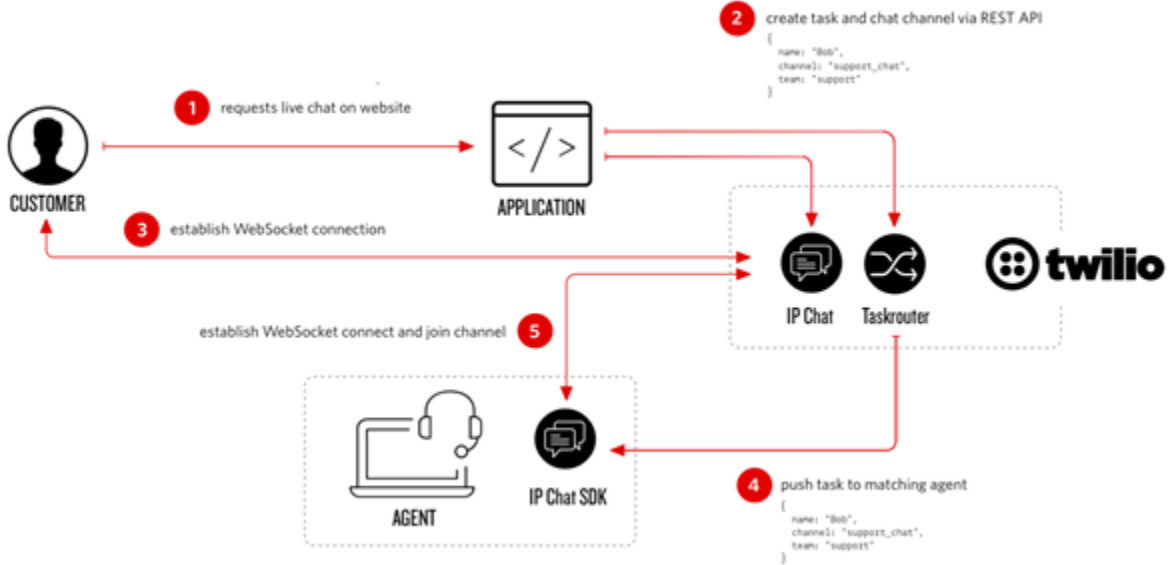
4.4.3 Web Chat:
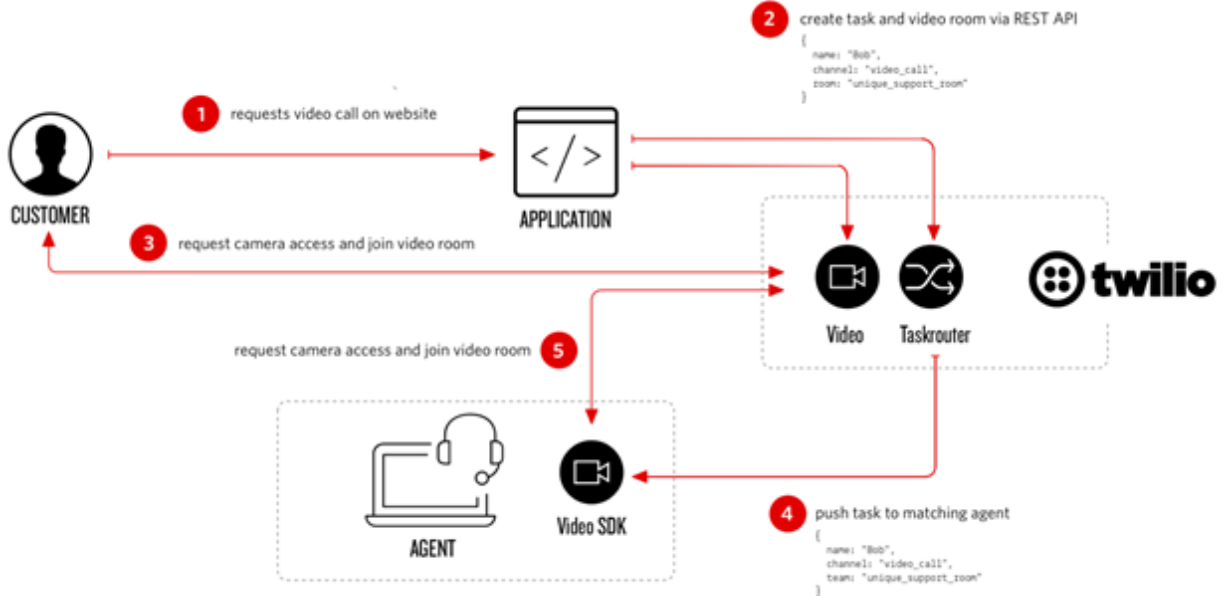

Figure 6 web chat

4.4.4 Video Call:

Figure 7 video call

4.5 Implementation

4.5.1 Administration Page

A basic administration interface to create and delete new call centre agents.

Figure 8 Call Center Admin Page

4.5.2 Workers Page

The working environment for agents. Agents need to open this page and log in.
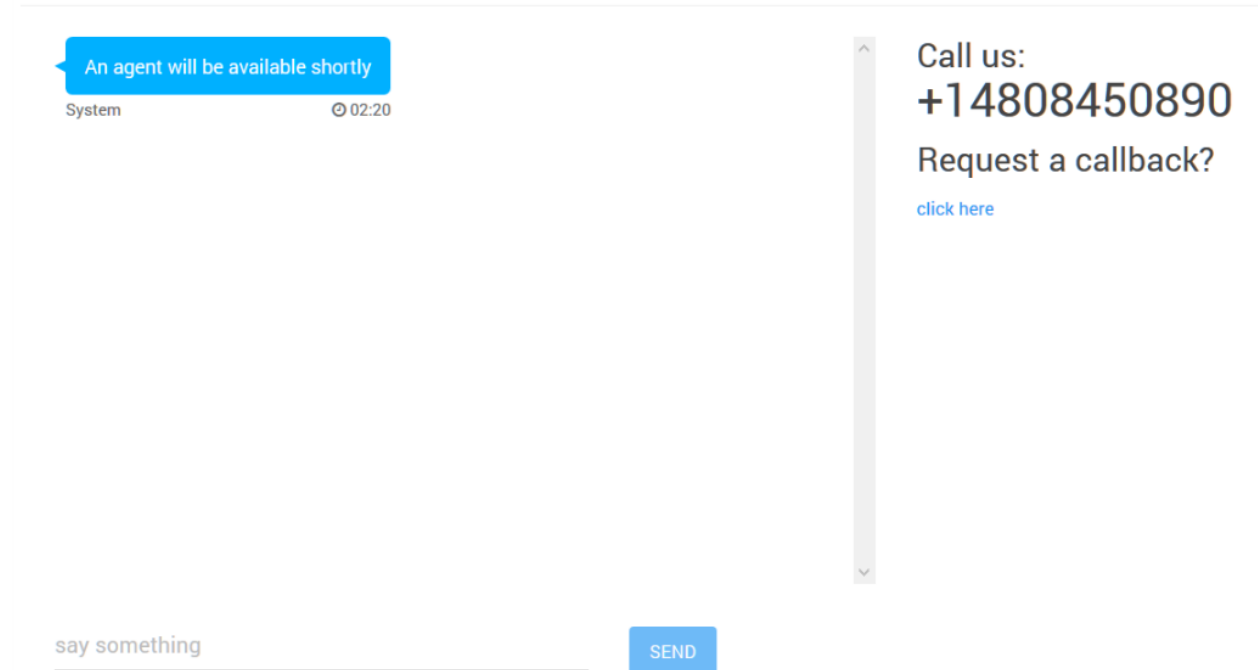
Figure 9 Call Center Workers Page

4.5.3 Customer Page
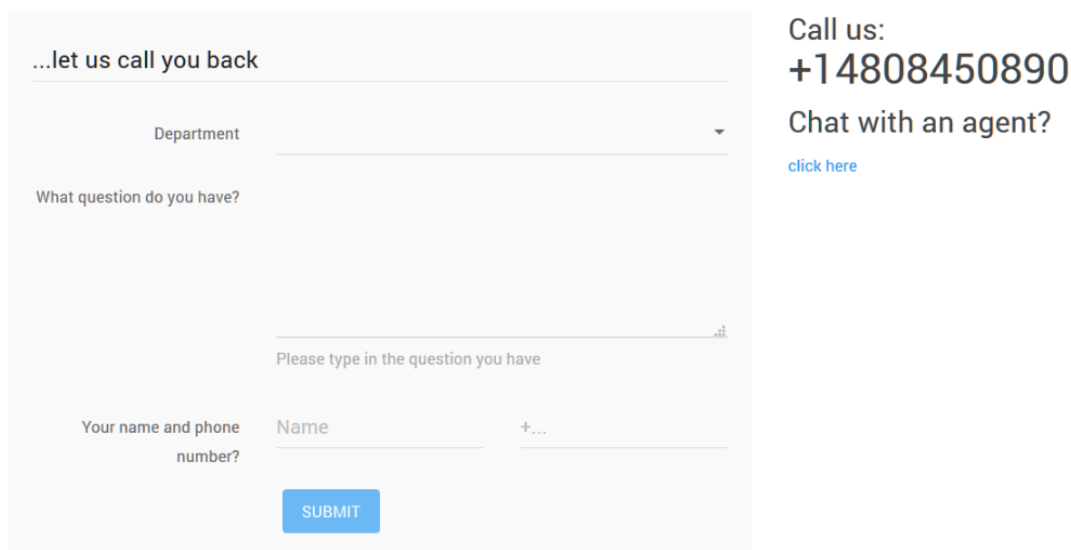
Twilio Contact Center: Live Chat

An agent will be available shortly

System   ⏱ 02:20

Call us:
+14808450890
Request a callback?

click here

say something   SEND

Figure 10 Call Center Customer Chat Page

Twilio Contact Center: Contact Form

...let us call you back

Department

What question do you have?

Please type in the question you have

Your name and phone
number?   Name   +...

SUBMIT

Call us:
+14808450890
Chat with an agent?

click here

Figure 11 Call Center Customer Call Page

Chapter 5: Conclusion

It's been a wonderful experience doing this project. We were able to learn Application Development in Python and Neural Network Concepts and feel this would be helpful in our future
endeavours. The field of machine learning and application designing is very wide with an unlimited options. We learnt few of these techniques and we look forward to utilizing this in future and learn more.

References

[1]A. Saeed, "Urban Sound Classification, Part 2", Aqibsaeed.github.io, 2017. [Online]. Available: http://aqibsaeed.github.io/2016-09-24-urban-sound-classification-part-2/. [Accessed: 24- Apr- 2017].

[2]"CS231n Convolutional Neural Networks for Visual Recognition", Cs231n.github.io, 2017. [Online]. Available: http://cs231n.github.io/convolutional-networks/. [Accessed: 24- Apr- 2017].

[3]2017. [Online]. Available: http://karol.piczak.com/papers/Piczak2015-ESC-ConvNet.pdf. [Accessed: 24- Apr- 2017].

[4]"Getting Started With TensorFlow | TensorFlow", *TensorFlow*, 2017. [Online]. Available: https://www.tensorflow.org/get_started/get_started. [Accessed: 24- Apr- 2017].

[5]"Keras Documentation", *Keras.io*, 2017. [Online]. Available: https://keras.io/. [Accessed: 24- Apr- 2017].

[6]A. Saeed, "Urban Sound Classification, Part 1", *Aqibsaeed.github.io*, 2017. [Online]. Available: https://aqibsaeed.github.io/2016-09-03-urban-sound-classification-part-1/. [Accessed: 24- Apr- 2017].

[7]"Twilio Docs", *Twilio.com*, 2017. [Online]. Available: https://www.twilio.com/docs/. [Accessed: 24- Apr- 2017].

[8]"Getting Started on Heroku | Heroku Dev Center", *Devcenter.heroku.com*, 2017. [Online]. Available: https://devcenter.heroku.com/start. [Accessed: 24- Apr- 2017].