REVIEW OF INTERNATIONAL GEOGRAPHICAL EDUCATION

ISSN: 2146-0353 ● © RIGEO ● 11(5), SPRING, 2021

www.rigeo.org Research Article

Design And Implementation IVR Outbound Service API Using Text-To-Speech

Esa Fauzi[1]
Department of Informatics Engineering, Faculty of Engineering Universitas Widyatama, Bandung, Indonesia
esa.fauzi@widyatama.ac.id

Adri Genta Rahdian[2]
Department of Informatics Engineering, Faculty of Engineering Universitas Widyatama, Bandung, Indonesia
adri.rahdian@widyatama.ac.id

Agustinus Ipan Suryana[3]
Department of Informatics Engineering, Faculty of Engineering Universitas Widyatama, Bandung, Indonesia
agustinus.suryana@widyatama.ac.id

Penta Al Ansyah[4]
Department of Informatics Engineering, Faculty of Engineering Universitas Widyatama, Bandung, Indonesia
al.ansyah@widyatama.ac.id

Tiara Nastiti Handana Ningtias[5]
Department of Informatics Engineering, Faculty of Engineering Universitas Widyatama, Bandung, Indonesia
tiara.handana@widyatama.ac.id

Kinanti Dara Nurkhofifah[6]
Department of Informatics Engineering, Faculty of Engineering Universitas Widyatama, Bandung, Indonesia
kinanti.dara@widyatama.ac.id

Abstract

In business world, customer support is one way to build relationships with customers. Phone calls are one of the customer support media that is widely used because of its flexibility. But the limited customer support workers can cause customers to wait a long time to be served. One solution that can be done is to install an IVR (Interactive voice response) system. IVR is a system mechanism that provides automated talks based on the response of DTMF (Dual-tone multi frequency).

In addition to serving customer complaints or inquiries, IVR can also be used to provide information to customers. This IVR is called the outbound IVR service because it is the companies that make phone calls to customers. In this study, an IVR system was created with outbound service model. The IVR system also provides a feature for companies to create and manage their own IVR menus by inputting response numbers and words that will be listened to customers. In the process, the system also uses a text-to-speech library to convert text to voice, using the PJSIP library to make phone calls over sip protocol, using MongoDB as database, and Redis as cache.

Keywords

IVR, IVR Outbound Service, Text-To-Speech.

To cite this article: Fauzi, E.; Rahdian, A, G.; Suryana, A, I.; Ansyah, P, A.; Ningtias, T, N, H.; and Nurkhofifah, K, D. (2021) Design and Implementation IVR Outbound Service API Using Text-To-Speech. *Review of International Geographical Education (RIGEO), 11*(5), 789-796. doi: 10.48047/rigeo.11.05.74

Submitted: 02-10-2020 ● **Revised:** 06-12-2020 ● **Accepted:** 12-02-2021

Introduction

Currently all business that service or product-oriented needs customer support services. The sole purpose of customer service is to meet customer expectations. One of these customer support service media is by using telephone calls center. The call center, originated from the civil airline industry, an application using telecommunications that is used by many industries to maintain important relationships with customers (Dai & Li, 2012). But these call centers are often constrained by the lack of workers in addressing phone calls. As a result, customers have to queue and wait longer. In fact, customer waiting time is the key to the quality of service (Liu, Gong, Ma, & Yu, 2017).
IVR or interactive voice response is a system that can be used to handle customer service. IVR can be a front-end before customers connect with customer support employees. IVR system becomes very important because it can handle 20-60% of phone calls in the call center system (Karademir & Heves, 2013). In the conventional IVR system model, the customer makes a first call to the call center telephone number then the caller navigates the menu using DTMF (dual-tone multi frequency) until the request is complete or continues to CSR (Customer Support Representative) if the request has not been completed (two stage IVR system) (Chavan & Dr.Y.V.Haribhakta, 2016).

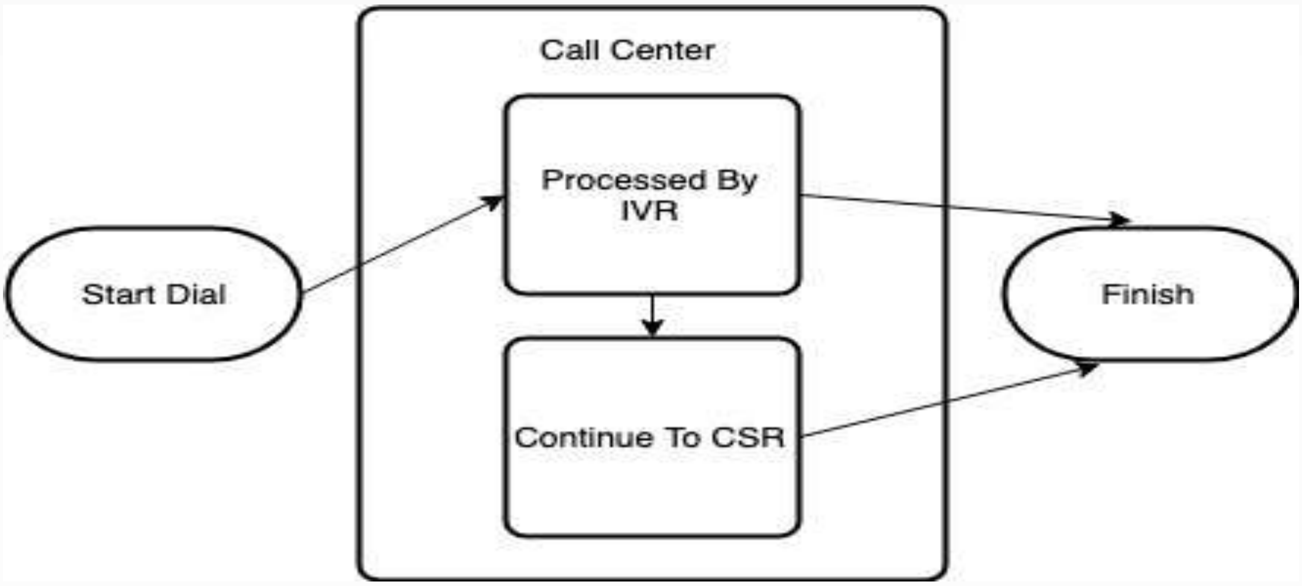

Fig. 1 Conventional IVR System

Unlike the conventional IVR system, in this paper an IVR system was developed using an API (Application Programming Interface) through SIP (Session Initiation Protocol) where calls will not be made by the customer but by the API. The advantage of this system is that it provides a way to create their own IVR, so that IVR service providers can determine their own menus in their IVR services.

Related Works

Many studies related to IVR have been carried out, but most of them are similar to conventional IVR systems, starting from the customer call and then to the IVR system until it is finished. Some of them are IVR using hidden markov model speech recognition system by Mohammed Hamidi where this research focuses on creating a secure Telephony Amazigh Spoken System over the network (Hamidi, Satori, Zealouk, Satori, & Laaidi, 2018). There is also IVR research using SIP (Session Initation Protocol), one of which is research by Ansari. Ansari also uses the open source free switch tool as a private branch exchange server (Ansari, Nehal, & Qadeer, 2013). But the difference with this study, in this study the system built can create its own menu for its IVR service.

Technology stack

DTMF

Dual-tone multi-frequency signaling (DTMF) is a telecommunication signaling system using the voice-frequency band over telephone lines (Dodd, 2012). This DTMF is used to get keypad input from the user.

SIP

To implement telephone calls over the internet one technology that can be used is the Session Initiate Protocol (SIP) (Rosenberg et al., 2002). SIP is a signal protocol on the internet that can be connected to traditional telephone networks or public switched telephone networks (PSTN).

Text To Speech

A text-to-speech (TTS) is a system that transform common language text into speech signal (Sawant & Borkar, 2018). In this study, text to speech will be used to change the language text to Indonesian or English

Redis

Remote Dictionary Server or Redis is an open source that used as cache, message broker and database. It support rich data structure and can handle about 100 thousand requests per second (S. Li, Jiang, & Shi, 2017). Redis is used to cache audio in this IVR system.

Mongo

Mongo DB is an open-source database system based on distributed files. Mongo DB is used as a data structure of the JSON binary format to manage and store data (C. Li & Yang, 2014). Mongo DB in this study is used to store queue requests.

Golang

Golang is an open-source programming language developed by Google Inc. The Golang Programming Language does not support object-orientation but offers packaging similar to classes. Some famous projects like docker and kubernetes use golang (Yasir, Asad, Galib, Ganguly, & Siddik, 2019).

PJSIP

PJSIP is an API written in the C programming language that implements many protocols such as SDP, SIP, STUN, RTP, ICE, and TURN. This API combines NAT traversal functionality and a rich multimedia framework into a high-level API. In this study PJSIP is used for calling the user (PJSIP-Open Source, 2014).

Architecture and Design

This research is designed to produce API and IVR system combined with Text-to-speech feature. The API is used as an interface for publishers. Publisher itself is a term for people who will create their own IVR menu. Then there is the IVR System itself. This IVR system will be combined with text to speech. The goal is to get the audio to be sent to the user. So broadly, this research is divided into 3 parts, namely API, IVR, and text-to-speech.

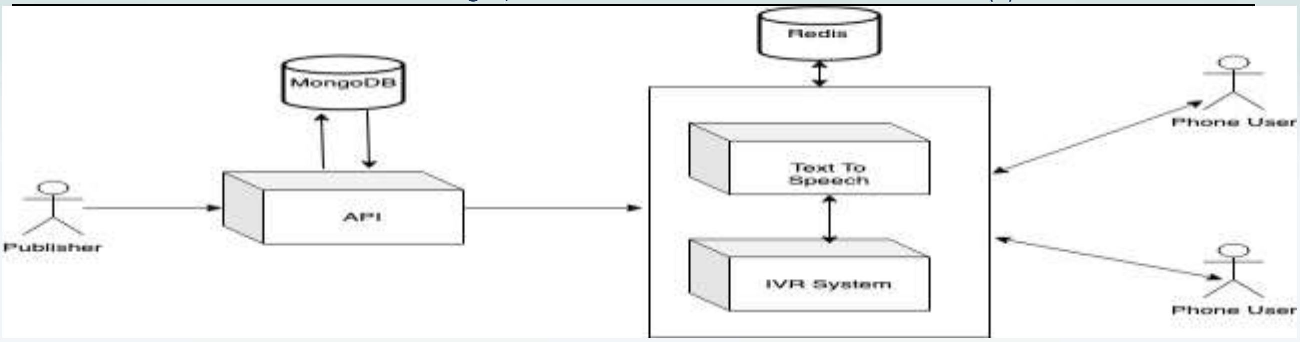

Fig. 2 Proposed Architecture IVR

API

The API will receive input from the publisher in the form of input numbers, IVR menus, and words that will be converted into audio. If the input is valid then the API will continue to call the IVR server. However, to prevent a race condition from occurring, before calling the IVR server (because the PJSIP is limited to 8 accounts when handling several simultaneous requests), the input data will be queued first using mongo DB. The algorithm to handle this race condition follows the algorithm created by Esa in his paper on voice OTP (Fauzi & Edison, 2020).

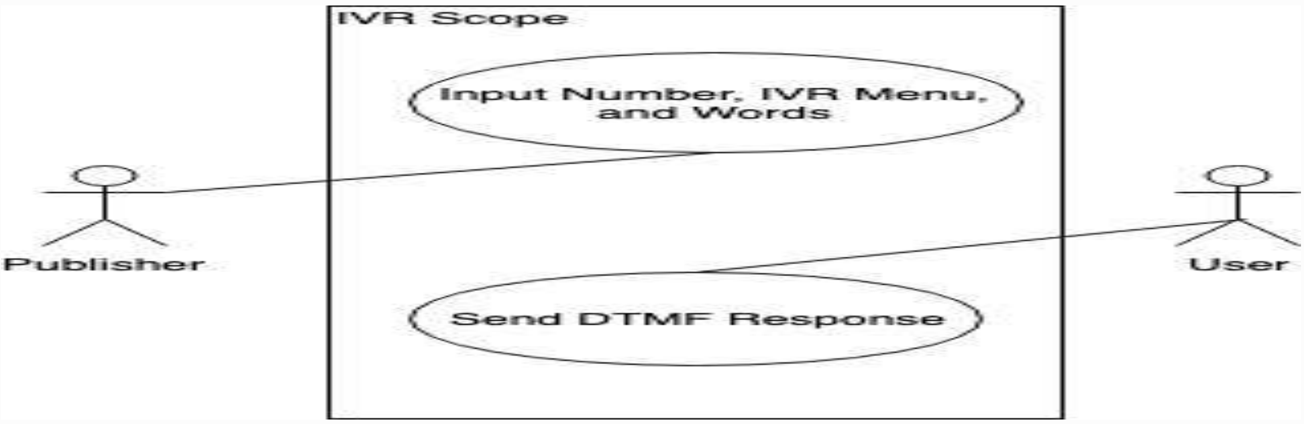

Fig. 3 Use Case IVR System

IVR

When the IVR server receives a request from the publisher, the IVR server via the SIP protocol will try to invite the target user. If the user picks up then the system will print audio using text-to-speech which will then be listened to the user. After listening to the audio, the user can respond using DTMF (by pressing the selection key on the phone). Then if after the DTMF response there is still a sub menu, the audio generation process will be carried out again and the user will return the DTMF response. This process will continue to repeat until it is finished. When finished, the system will send the user's DTMF data to the publisher's server as a callback.

Text-To-Speech

In this research, the text-to-speech system will use Redis as the audio cache. So, the audio that has been generated will not be reprinted. The pattern is to store hashing text on each audio that has been generated. So before printing audio the text-to-speech system will check the hashing of the existing text. If it is present then the audio will be printed, but if it is not there the system will fetch the existing audio.

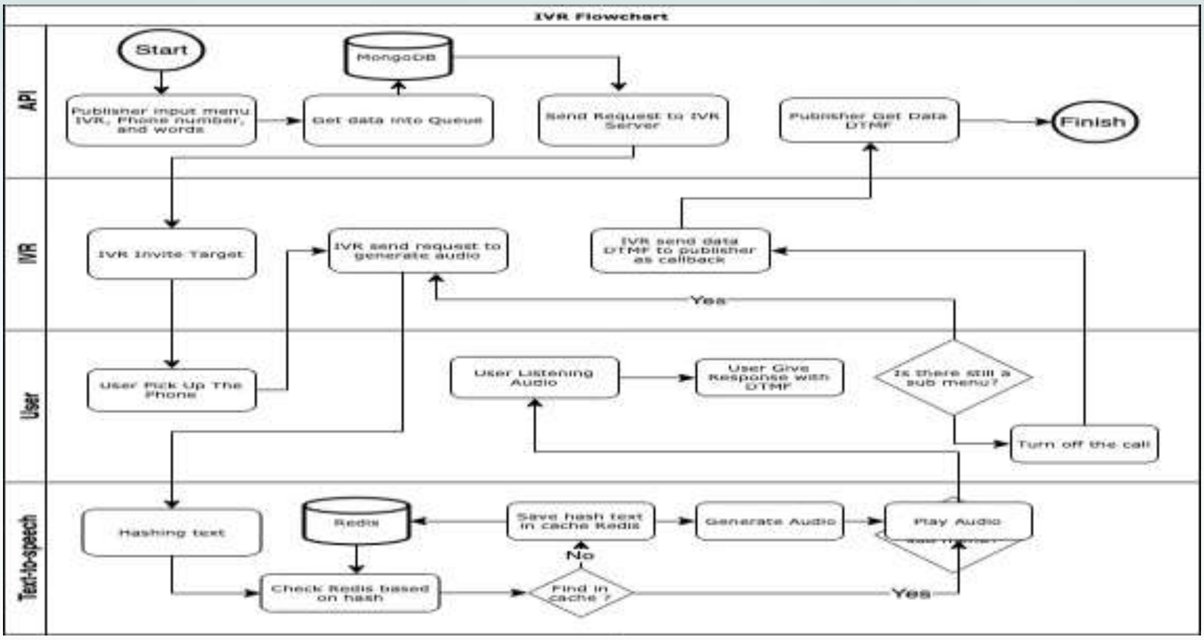

Fig. 4 IVR Flowchart

Implementation

The IVR system API is divided into 2 parts, namely IVR menu entry and callbacks to get user DTMF data. In the IVR menu entry, all data is entered into JSON, here is an example:

Example JSON Entry IVR Menu
Path: /ivr

```
curl -i -X POST http://exampleivr.id/ivr -d '{
 "Subscriber_id": "developertampan",
 "Intro_text": "Welcome to service IVR.",
 "Closing_text": "Thank You",
 "Dtmf_timeout_text": "Sorry, you did not enter an option. This call will be terminated.",
 "Unknown_action_text": "Sorry, the option you entered was not recognized. This call will be terminated.",
 "options": [
  {
   "key": "1",
   "text": " For ABC services, press one.",
   "action": "finish"
  },
  {
      "key": "2",
      "text": "For a crash report, press two.",
      "action": "next",
      "suboptions": [
       {
        "key": "1",
        "text": " For an SMS OTP service interruption report, press one.",
        "action": "finish"
       },
       {
        "key": "2",
        "text": "For the SMS Notification service interruption report, press two.",
        "action": "finish"
       }
```

```
      ]
    }
  ]
  "callback": "http://mockbin.org/request",
  "callee_number": "08999xxxxxx"
}
```

Here's an explanation of each field:

a. subscriber: the id of the subscriber using the IVR API
b. intro_text: text that is read when you pick up the phone for the first time
c. closing text: text read at the end of the IVR
d. dtmf_timeout_text: text that is read out if the user does not respond to DTMF input
e. unknown_action_text: text that is read when the user inputs an incorrect number
f. options: field to store IVR menu, type array of object. This options field contains the following fields:key: nomor menu IVR

1. text: text that is read out for a certain key
2. action: contains the status of the IVR menu. If the last input "finish". If it still continues, enter "next". If you return to the previous IVR menu, enter "back". If calling to a certain number, enter "call_to".
3. call_to: if the action contains "call_to", then fill this field with the destination number.
4. suboptions: contains data such as options, but is intended for sub options of certain keys. This field is required if the action field is "next".

g. callback: url or callback path
h. callee_number: destination phone number

This IVR system has a callback function that can be called to get the user's DTMF data. The following is an example of this callback:

Example: callback

```
POST /your-callback-route
Host: http://your.callback.com
Content-Type: application/json
{
   "target": "08999xxxxxx",
   "result": "1,2,3"
}
```

There are 2 parameter fields in the callback function, namely:

a. target: the number that has been called
b. result: the number the user has selected

However, if the user inputs incorrectly, the callback function will return an error. Here's an example:

Example: callback error

```
POST /your-callback-route
Host: http://your.callback.com
Content-Type: application/json
{
  "target": "08999xxxxxx",
  "error code": "INV-0001",
  "error": "Callee missed the call"
}
```

The error code field provides the error code of the error and the error field provides an explanation of the error.

Conclusion

IVR system is a system that can help companies as a customer support tool. In this study, the IVR system that was built can help companies to make their own IVR because in this system the company can define what menus they want to listen to and the response number. The use of the queuing algorithm in using the IVR server itself can prevent race conditions from occurring because requests to the server are arranged in queues. In addition, the use of redis can reduce the burden on the server in generating audio because the same audio is not reprinted.

References

Ansari, A. M., Nehal, M. F., & Qadeer, M. A. (2013, 26-28 July 2013). *SIP-based Interactive Voice Response System using FreeSwitch EPBX*. Paper presented at the 2013 Tenth International Conference on Wireless and Optical Communications Networks (WOCN). doi:https://doi.org/10.1109/WOCN.2013.6616224

Chavan, S., & Dr.Y.V.Haribhakta. (2016). IVR-O-Fly : IVR Configuration On Fly. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 5*(4), 1169-1173. Retrieved from http://ijarcet.org/wp-content/uploads/IJARCET-VOL-5-ISSUE-4-1169-1173.pdf

Dai, T., & Li, J. (2012). *A simple method on staff scheduling in multi-skill call center*. Paper presented at the ICSSSM12. doi:https://doi.org/10.1109/ICSSSM.2012.6252201

Dodd, A. Z. (2012). *The Essential Guide to Telecommunications* (5th Edition ed.): Prentice Hall. Retrieved from http://www.pce-fet.com/common/library/books/51/9852_TheEssentialGuidetoTelecommunications5thEdition2012.pdf

Fauzi, E., & Edison, A. (2020). ARCHITECTURE OF ONE TIME PASSWORD WITH VOICE CALL FROM SIP TO PSTN. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 24*(2). Retrieved from http://repository.widyatama.ac.id/xmlui/handle/123456789/12219

Hamidi, M., Satori, H., Zealouk, O., Satori, K., & Laaidi, N. (2018, 30-31 Oct. 2018). *Interactive Voice Response Server Voice Network Administration Using Hidden Markov Model Speech Recognition System*. Paper presented at the 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4). doi:https://doi.org/10.1109/WorldS4.2018.8611591

Karademir, R., & Heves, E. (2013, 1-4 July 2013). *Dynamic interactive voice response (IVR) platform*. Paper presented at the Eurocon 2013. doi:https://doi.org/10.1109/EUROCON.2013.6624972

Li, C., & Yang, W. (2014). *The distributed storage strategy research of remote sensing image based on Mongo DB*. Paper presented at the 2014 Third International Workshop on Earth Observation and Remote Sensing Applications (EORSA). doi:https://doi.org/10.1109/EORSA.2014.6927858

Li, S., Jiang, H., & Shi, M. (2017). *Redis-based web server cluster session maintaining technology*. Paper presented at the 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). doi:https://doi.org/10.1109/FSKD.2017.8393274

Liu, S.-A., Gong, J., Ma, L., & Yu, M. (2017). *Influence of waiting times on customer loyalty and queueing behavior in call centers*. Paper presented at the 2017 29th Chinese Control And Decision Conference (CCDC). doi:https://doi.org/10.1109/CCDC.2017.7978688

PJSIP-Open Source, S. (2014). Media, and NAT traversal library. Retrieved from https://trac.pjsip.org/repos

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., . . . Schooler, E. (2002). *RFC3261: SIP: session initiation protocol*. RFC Editor, Retrieved from https://dl.acm.org/doi/pdf/10.17487/RFC3261

Sawant, N. K., & Borkar, S. (2018). *Devanagari Printed Text to Speech Conversion using OCR*. Paper presented at the 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on. doi:https://doi.org/10.1109/I-SMAC.2018.8653685

Yasir, R. M., Asad, M., Galib, A. H., Ganguly, K. K., & Siddik, M. S. (2019). *Godexpo: an automated god structure detection tool for golang.* Paper presented at the 2019 IEEE/ACM 3rd International Workshop on Refactoring (IWoR). doi:https://doi.org/10.1109/IWoR.2019.00016