

Chatbot Using Dialogflow and Web Services

Kannadasan R, Prabakaran N, Krishnamoorthy A, Nerella Tarak Krishna, Sai Vishnu Giri, Konapalli Pavan Kumar Reddy

Abstract: *The Chatbots are helpful in making human effort minimal for less-complex tasks like answering messages and also to provide great satisfaction to the clients round the clock. This is considered a better initiative as the business organizers don't have to worry about reaching to customers. There are many proposed models to get the desired output most of these are helpful if the user input is always from the given set of words the software is designed for, but unfortunately, this is not the scenario always as the user shall have a different way to message. This and the other models are being replaced by the Artificial Intelligence systems which have the capability to identify the structure and words from a given context and use it predict the best possible outcomes. In our proposed model, we are using the same and it is handled by Dialogflow, which helps in achieving an end-to-end bot without the user being worried about the algorithm that should be used to train this bot. Using a configured server with Dialog flow helps in handling various requests. Using technologies like Dialog flow, Node.js in this proposed model, an effort is made to make better bots with better functionality and continuation in the conversation. The system here is proposed to analyse the user query and respond back to the user with an appropriate answer. Our model is developed to help user in searching restaurants around him with different cuisines, categories else to see the list of top reviews of a hotel and it's cuisines. This chat-bot since being location dependent, it shall be using the location of the user to fetch real-time results for the queries put forward by the user.*

Keywords: Chatbot, Node, Dialogflow, Machine Learning.

I. INTRODUCTION

Chatbots helpful in a lot of ways. They help in sustaining a business with great of profits as the customer is always maximum. Chatbots are intelligent software programs that are returned to answer user queries which are specific to the purpose they serve. Using these bots, the customer is always satisfied as he gets the information that he needs irrespective of any human intervention. These chatbots help in establishing a new environment in having between user and

Revised Manuscript Received on November 27, 2019.

* Correspondence Author

Kannadasan R, Vellore Institute of Technology Assistant Professor (Senior) E-mail id :kannadasan.r@vit.ac.in

Prabakaran N, Vellore Institute of Technology Assistant Professor (Senior)E-mail id :prabakaran.n@vit.ac.in

Krishnamoorthy A, Vellore Institute of Technology Assistant Professor (Senior) E-mail id :krishnamoorthy.arasu@vit.ac.in

Nerella Tarak Krishna, Vellore Institute of Technology, Mail Id: ntk05071998@gmail.com

Sai Vishnu Giri, Vellore Institute of Technology, Mail Id: sai344@gmail.com

Konapalli Pavan Kumar Reddy, Vellore Institute of Technology, Mail Id: pkreddy430@gmail.com

system. Instead of a traditional interface which makes us feel that the interaction is with the system, the chatbot provides a better way of interaction which makes the user feel as if the conversation is happening with another human. For the same reason these are called Virtual Agents, but, we can also call it as a Personal digital assistant.

The world that we live in now is an era where everything is available and made possible using the internet. If we have basic resources like internet, we can get all type of information, but this can be made possible by typing the same request to chatbots and thus causing their invention. Depending on the service that should be served like customer services, shopping, food order, news updates, reservation etc to the user, we can develop the chatbot. The bots can handle multiple requests from various services too, but they must be developed in this way. The popularity of messaging apps suggests people feel happy talking to chatbots and getting their requests served, as the bots provide the required information in an interface which shall make the user feel like that the user is speaking with another user. Basically, chatbot requires chat interface or chat window which takes user inputs and provide a response message to the user and hence making it feel like a normal conversation to the user.

The motivation behind making chatbots is to make less human effort for less-complex tasks like answering messages and to provide great satisfaction to the clients. This is considered a better initiative as the business organizers doesn't have to worry about reaching to customers and updating them every time. This script which is written once and managed occasionally can make a huge contribution to the company by focusing the developers and other staff on things that needs their attention.

II. LITERATURE SURVEY

A. Intelligent Chat Bot for Banking System (IJRASET):

This model proposed a noble effort in making an intelligent chatbot which is built using technologies like C# and .NET. In here, the user query is interpreted using these technologies and the voice to text integration is also done using the same technology. The main of this model is to propose a model which can accept the queries from the user in both text and audio format. This helps in easing the input received from the user and makes the user feel more comfortable in speaking with the bot. The core language libraries are used to convert the voice to speech.

Using Artificial Intelligent Mark-up Language (AIML) is used in this model to analyse the patterns of the strings that are



given by the user and shall be responding back to the user depending upon the same criteria. The Artificial Intelligence (AI) is made available to the user by the user's message being processed using a third-party expert system. This helps in achieving the output without general string comparisons, but with some training on the specific contexts and structures of sentences. The response generated from this is sent to the user and shall accept another query from the user and performs the same actions.

B. College Chat-bot (IJARCET):

This model put forwards a proposal in building a chatbot using artificial intelligence algorithms which analyses and understands user's queries and messages. This is a web application that provides answer to the query of a certain set of predefined customer base of students. These clients are expected to query the bot for chatting. Java is used, and the application is available in web, mobile and desktop environments. The users can chat using any format there is no specific format the user has to follow. This application is built using APL.ai and serves the user by accepting the queries and forwarding it to APL.ai and sending the response back to the user from APL.ai.

Interactive Question-Answering is explained in this model which states that the Interactive question-answering systems turn the focus to the interaction between the user and the program, not just the question-answering. And by adding more weightage to the previous point, it is states that these systems allow either the user or the system to play a greater role in the conversation by making the system understand what the specific interests of the user are and suggesting materials which are related or even refinements to a user's query respectively. It is stated that an interactive layer in a system gives rise to more complex queries that can be issued by the user as mistakes made by the chatbot can be rectified through user message.

The above specified models with many other specify a great deal of advantages by serving the users in a better conversational way rather than general system-user interaction. There are few aspects that are considered and improvised in the model this paper put forth. Firstly, the request is either sent to the third-party Artificial Intelligence system based directly else by using the available integrations available. The Dialogflow technology is being handled by Google, which has developed its APIs with authentication to use different bots and to send and receive requests and responses from them accordingly. This response, however might not be complete as few times the user would like to search for specific and in-details specifications about something, in that scenario our system must be intelligent enough to understand what the user is saying and accordingly it must contact the outside world on this basis and must be able to take a wise decision in doing so according to the user given query.

When the user uses the voice recognition the bot must be able to identify the message correctly and for this job Google is the best technology available as it has around 95% accuracy [4] and the DialogFlow uses the cloud APIs [3] which achieve this for taking input from user.

Dialogflow [5] also offers many default use case scenarios like that of entities list, fallback intents, small talks .,etc, which makes the model more easy and hassle free. The chatbot must always be available and hence, trusting it with the Google's infrastructure is the best option as it is designed to handle almost all cases of faults and failures of its servers and also implements load balancing to not to increase the load on a single server [6]. With these options we can also make sure that the response when being sent from the Dialogflow to the user, the server can interpret the response by adding more or modifying the whole response. There is an alternate way to utilize this server by deleting the contexts if there are any unnecessary contexts that are active. This can be performed using API call to the Dialogflow to delete the specified contexts.

III. PROPOSED WORK

A. Aim

To perform tasks that can be done without human intervention, if the company uses its human resources it might cost more rather than software which understands and interpret the given questions and answers it accordingly. Using the proposed architecture, the user shall be served with different set of restaurants based on his real-time location and this is made possible using constructive system architecture with an intelligent bot behind it to handle various scenarios and act give the response accordingly. For few of the scenarios, the bot shall be able to contact different services, fetch the required results and display it to the user. The better working of the bot depends upon the proper wireframe of the conversational flow and the relevance of the answers it provides for the given question. In our model, this is well thought and only the required information is being displayed to the user and remaining part is ignored.

B. Objectives

In the proposed model we have developed in chatbot which takes queries from user interpret the meaning inside it and performs certain tasks like viewing the restaurants around the user checking for a specific cuisine or to see the reviews of a Hotel. This is achieved by using Google's dialogflow which provides us with the functionality to create our own chatbot and have a server running at the background with uses cases of performing certain actions depending upon the query given by the user. This chat-bot since being location dependent, it shall be using the location of the user to fetch real-time results for the queries put forward by the user. The relevance of the answers given to the user defines the functionality of the bot and the accessibility to various options provides a better way for the user to interact with the bot. The cautions are taken that the bot shall be responding irrespective of the questions asked by users and shall respond with relevance in its content.

IV. FRAMEWORK, ARCHITECTURE OR MODULE FOR THE PROPOSED SYSTEM

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves



Careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

A. MODULES

1. Receiving the user's query.
2. Contacting DialogFlow with query.
3. Interpreting DialogFlow's response
4. Contacting Zomato APIs.
5. Returning the response back to user.

1) Receiving the user's query: Using the developed botUI's user interface the user gives the query. This query can be received by botUI and is sent to server using AJAX request. Since the user interface and the backend lies in the same server, there is no usage of CORS (Cross-Origin Resource Sharing) headers since the resources are shared within the same server. The botUI when sends the AJAX request using JQuery and keeps the adds a message with 3 loading dots which specifies that the response for the given query is being processed and shall be displayed as soon as it receives it. The request needs to be sent to a server which sends the same query to the Dialogflow using the APIs provided with the basic authentication provided in the authorization headers which consists of the clients access token which is available in the console of the Dialogflow.

2) *Contacting DialogFlow with query*: The request received from the user is sent to DialogFlow using Node.js request npm package. The conversations in Dialogflow are handled using session-ids and based upon the session-ids it saves the information and uses it accordingly with every subsequent request. The DialogFlow analyzes the given sentence and there are 3 routes that are possible in this scenario.

a) Dialogflow matches the given query with one of the intents whose in-context lies in the active contexts of that conversation. In this case, the intent which has the sentence it requires is fired and the entities are recognised. If all the required entities are not present then the Dialogflow prompts a message to the user to enter the respective field and till the user enters this field else makes another query, the Dialogflow shall not take the request forward and keeps prompting the same message. If all the required parameters are given in the Dialogflow then it sends back the intent name, context name(out-context) of the activated intent, entities identified with the default response, if given any. If the given request has to get the reviews for the specified hotel, the JSON which we shall receive is different and accordingly we can call Zomato APIs for reviews and send the info to the botUI which shall display the reviews in an orderly fashion.

b) Dialogflow doesn't identify the given query from the user and under this scenario, the general response we get can be configured in the default fallback intent that is specified for each of the chatbot. The Dialogflow selects one of these default messages given in that intent and gives out response whose intent states default fallback intent. Here, we can have a specific default fallback intent for each

of the intents and these intents gets fired if its previous intent was active and the user typed something that the chatbot is not trained for.

c) Dialogflow's small talk can be triggered using some of the predefined statements, this increases the interactivity of the bot by making the users to speak generally with the bot without any specific intentions.

3) Interpreting DialogFlow's response: After receiving the response from the dialogflow, depending upon the entities that are recognised, the calls are made to Zomato APIs. First the location-id is identified for the given city name and this location id is used in all of the subsequent requests made to Zomato. The other requests depends upon the intents and the entities identified. There are developer-defined entities in Dialogflow named `general_hotel_category`, `hotel_categories`, `cuisines`. These entities are very useful as these helps us in identifying the requests of different users irrespective of the vocabulary they use.

a) `general_hotel_category` has the list of variables used to specify general hotel categories like hotels, restaurants, etc.

These doesn't have a fixed set of cuisine else a category of ambience.

b) `hotel_categories` has the list of variables that can be used to specify a specific hotel category which are predefined in Zomato. It can be helpful when user sends dinner, supper, main meal to specify that user is looking for dinner.

c) `Cuisines` has the list of different cuisines that are available in Zomato which are obtained using multiple calls to the category APIs of Zomato.

The existing 2 intents to handle these cuisines `cuisines_specific_places_1`, `general_and_category_specific_places_1` which are to identify the requests with general hotels input or with some category like cafe etc and the other one to identify the cuisine based queries respectively. Both of these intents have around 25-30 user says using which the bot is trained for and these sentences are trained in the bot using machine learning algorithms. We can also choose to not to use apply any machine learning algorithms in training the bots in that scenario the bots shall work with the basic string matching algorithms with the given user query.

4) *Contacting Zomato APIs*: After detecting the intent that is triggered and depending upon the entities that are present in the query the API calls are made. The whole set of APIs are

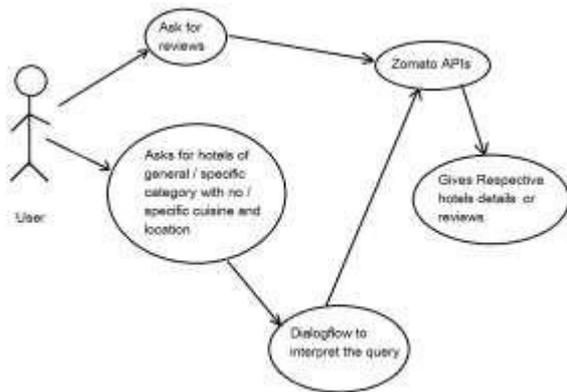
- a) General hotels in a specific location. Example: Hotels in Chennai.
- b) Category based hotels like cafe/pubs at a specific location. Example: Cafes in Hyderabad.
- c) A specific cuisine at a given location.

Example: Italian cuisine in Mumbai.

- d) A certain cuisine and hotel category type. Example: Chinese cuisines in Chennai.

Since these have cuisines or hotel category involved in their requests, the Zomato APIs cannot understand these texts, so we maintain a database which has a list of these cuisine names and their associated ids assigned by Zomato. Before contacting the Zomato APIs we get the ids of the respective cuisine or the category and accordingly we shall query these hotels and get the respective output. This output is formatted into a JSON with the fields we shall need like hotel name, cuisines available in that hotel, hotel image, hotel id and aggregate rating.

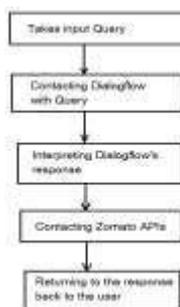
5) Returning the response back to user: From the given output from the server, the JSON is interpreted and displayed as cards in botUI. The buttons are created with the hotel id which shall send back to the server which shall use it for displaying the reviews. The format in which this is sent is different compared to a normal query which makes



it easy for the server to understand to not to send the query to the dialogflow, but instead to use the id sent to get the reviews of that hotel and display the same to the user. As how an individual request is sent using AJAX in the same way the button click is also handled using AJAX call.

B. Proposed System Model (ER Diagram/UML Diagram/Mathematical Modeling)

- 1) System Architecture:



- 4) Dataflow:

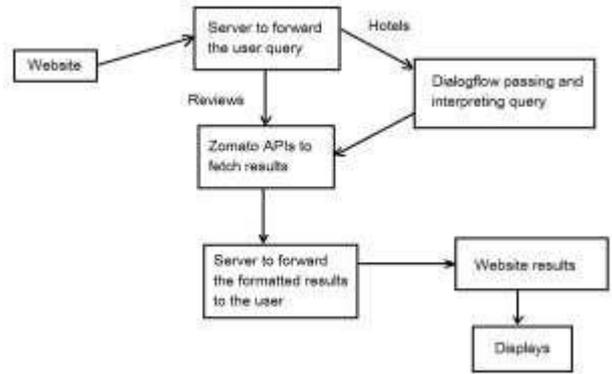


Fig.4 The above figure represents dataflow of the model

V. PROPOSED SYSTEM ANALYSIS AND DESIGN (AS PER IEEE STANDARD)

Fig.1 Represents the system architecture of our model.

- 2) Usecase diagram:

Fig.2 Represents the user case diagram for our model

- 3) Activity Diagram:

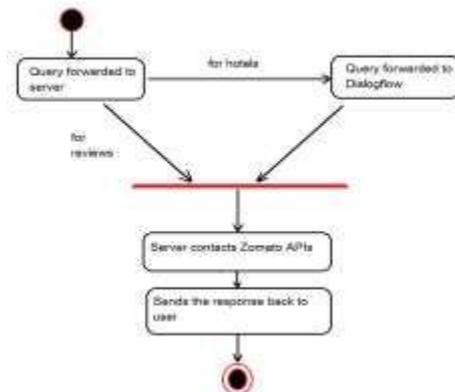


Fig.3 The above diagram represents the activities of our model

A. Introduction

The proposed system is analyzed depending on the user requirements and a feasibility study is done based on the given requirements and it has been found that the system is feasible to be developed. The user requirement is that the chatbot takes queries from user interpret the meaning inside it and performs certain tasks like viewing the restaurants around the user checking for a specific cuisine or to see the reviews of a Hotel.

B. Requirement Analysis

- 1) Functional Requirements:

a) Product Perspective: The Product perspective is to serve the clients/users in a better way by developing a chatbot which takes queries from user interpret the meaning inside it and performs certain tasks like viewing the restaurants around the user checking for a specific cuisine or

to see the reviews of a Hotel. Performing this task shall make the company to use its human resources and it might cost more rather than a chatbot which understands and interpret the given questions and answers it accordingly. This maximizes the revenues by enlarging the company's outreach irrespective of time and place and the availability of humans.

b) Product features: The chatbot is trained to have a good chat with user, for which it responds by displaying a list of hotels to questions like

1. General hotels in a specific location.
2. Category based hotels like cafe/pubs at a specific location.
3. A specific cuisine at a given location.
4. A certain cuisine and hotel category type.
5. Visit the hotel's site.
6. View reviews for the hotels given.
7. Displaying the other available cuisines for that hotel.

The above results are fetched using Zomato APIs under developers account.

c) User characteristics: This software is built on the basis that the user shall use it in the specific manner that was thought

of while designing and shall understand the features and limitations of his role and that of the bot.

d) Assumption & Dependencies: Since the software deals with realtime data and needs to have access to our servers, reliable internet connection and a browser(better if used Google Chrome) is assumed defaultly that the user shall be having to access this chatbot. There are several software dependencies that are used in creating the project and these must be maintained in the same version to avoid the conflicts between different versions of the same tool.

e) Domain Requirements: The software requires the APIs provided by Zomato must always be live as the bot relies upon it to get information for the user query. The availability of the other servers and softwares used like PostgreSQL, Node is also considered as an important requirement. The running servers with the proper functionality of the protocols underneath it is also considered in here as a requirement for the proper functionality of the bot.

f) User Requirements: The user is expected to have basic understands the basic operations of the a computer/laptop and has a browser(better if Google Chrome is used) installed in it. The project is developed based on the assumption that the user is familiar with usage of browser like opening a browser and clicking buttons and the effects that follows after clicking.

2) Non Functional Requirements:

a) Product Requirements:

- Efficiency- The Node.js technology used is single threaded event-driven, so the functionality of the code depends upon the events that are caused and because of the single threaded nature of Node.js, there will be not be any forking of processes else multiple threads working on different tasks. The Express framework is used in developing the APIs

in Node.js and using this efficiently reduces the number of lines that we shall have to code to achieve the desired output.

- Reliability- As all the technologies are well established and have a great developer support and are open source, the reliability on the code base can be termed as optimal. The other requirements would be to consider the working server to make the APIs be available for everyone at any specific time without being stopped by the server's power or operating system issues.
 - Portability- The application is developed to work in a browser. So, any popular browsers shall be supporting the app and since it is developed as a website, an embedding can be created for the same using which we can port the chatbot to different sites. The botUI npm package used in this project makes the bot to have responsive user interface. Hence, using this chatbot in a mobile also gives the same feel as one uses it in laptop. Even when developing the same bot for the mobile application, developer can make utilize of WebView and use this chatbot either from where it is deployed else by using a embedding.
 - Usability- The main of the chatbot is to provide with the user with a list of restaurants around him according to the user's specific requirements. The chatbot helps in achieving this task and also provides additional features like to see all the cuisines available in that hotel and to see the top 5 reviews of users for that restaurant. Since the chatbot works depending upon the current location of the user, these suggestions shall vary depending upon the location from where the user is accessing the chatbot.
- b) Engineering Standard Requirements:
- Economic- The economic standards for this chatbot revolves around the cost of maintaining the server. If the user traffic increases else if the number of Zomato API calls exceed 1000/day, then the partnership must be established with Zomato APIs which might add up for the economic model. If the other technologies used become paid service, then there might be increase in the economic model.
 - Environmental- The heat produced by the server might be causing a small issue with the environment if the infrastructure is huge, but since the operating system is designed to save power and run optimally with using minimal resources. The coolants and other chemicals can be used to reduce the heat and other by-products derived from the system.
 - Social- Socially, this chatbot provides a great use case as it helps the user to see the hotels and restaurants around. It provides the user to search for a place with the requirement he requires which increases the individual's socialization and interaction skills.
 - Political- The least concerns are of political as the project is completely based upon the autonomous individual software's and industries, and since there shall not be any interaction between the users the scope of internal politics is also greatly reduced.

- Ethical- This software is constructed and is use by a keen consideration of ethics. All the software used in this chatbot development are under proper terms in maintaining the ethics of the basic component. The interaction between the user and the system is limited to a certain extent by serving the requests made by the user. This provides a great way to overcome several ethical issues that can be caused.
- Health and Safety- Since, the results are fetched from Zomato's APIs and as the Zomato is one of the leading company that deals with the best hotels and cuisines, and there are user reviews to assist the user more about the hotel and hence the health and safety measures are taken care of.
- Sustainability- The software with its components are designed to maintain the sustainability in performance, cost and the minimal resources used. This is achieved by developing the chatbot using the efficient and reliable technologies.
- Legality- All the technologies used in this developed are under proper license agreements which allows a developer to modify it, develop using it else change the existing wireframe. The Zomato APIs are contacted using a proper authentication mechanism provided by them and hence making it legal to use their APIs and the technology that is built on it.
- Inspectability- As the codes are written in a modular basis using different frameworks and technologies, the inspectibility can be achieved every easy on component basis. This helps in troubleshooting an issue at a faster phase and more efficiently. As each of the component has its own specific role in contributing to the output we expect, when there's an issue else to test the system it becomes easy.

3) System Requirements:

a) Hardware Requirements:

- System Architecture: Intel®Core™ I3 CPU@ 2.30GHz
- Hard Disk: >=30BG
- RAM capacity: >=512MB

b) Software Requirements:

- Operating System : Ubuntu 16.04
- Technology : Node.js, PostgreSQL
- Frameworks of Node : Express
- Npm-packages : bluebird(^3.5.1),body-parser(^1.18.2), express(^4.16.2), nodemon(^1.15.0), pg-promise(^8.1.0), request(^2.83.0), botUI
- IDE : Sublime Text/ Notepad++

In order to achieve this output the system is built over different technologies like DialogFlow, Zomato APIs (Application Programmable Interface), Node.js, PostgreSQL,

JSON. As software is developed to execute few tasks which when compared to the performance of humans for the same, it should be more efficient and fast. Chatbots are pieces of software which are developed to automate human interaction with the clients. As it would cost company more to have a human sitting around waiting for the queries from the customers, if there is a software that is specially designed to handle this task the humans can work on handling more Complex tasks and this helps in satisfying the customer needs as well as to reduce other expenses for the company in having most human resources.

VI. RESULTS AND DISCUSSIONS

In this project, we have trained a bot and configure a server to take messages from the user and interpret it and gives back a relevant reply based upon the context and intentions of the user. The bot contacts Zomato using its APIs to get additional information to serve the user for the queries based on the hotels, cuisines., etc. It helps used in getting details like list of hotels with its names, cuisines, picture of the hotel, link to the respective hotel and an option to see the top reviews for the hotel. The bot is deployed in a Virtual private server and can be accessed in any browser (Google Chrome/Mozilla Firefox/Safari). This interface is connected with the bot internally using the server on which it is deployed. The requests are redirected to the bot in Dialogflow and the results are displayed in the browser using botUI in a formatted manner. For the testing purposes the site is deployed at www.blackwatersproduction.com and the URL of the chatbot is www.blackwatersproduction.com/botui/botui.html

A. SAMPLE TEST CASES

The sample test cases seen for the bot are as follows. The user types in the queries as follows commands:

a) Welcome message from bot

The bot must be able to comprehend every request given from user and should be able to start with the pre-configured welcome message.

b) Italian cuisines in Hyderabad

This query specifies the cuisines type and location name. This is given to the bot for processing.

c) Clicking on the read reviews button

This query gives the outlook on how the reviews are obtained for different hotels using the chatbot.

d) Chinese cuisines around me

The location is not specified in here to fool the bot by asking it to process this request.

e) Clicking on the image in the views

This is performed to see the hotel page in Zomato.

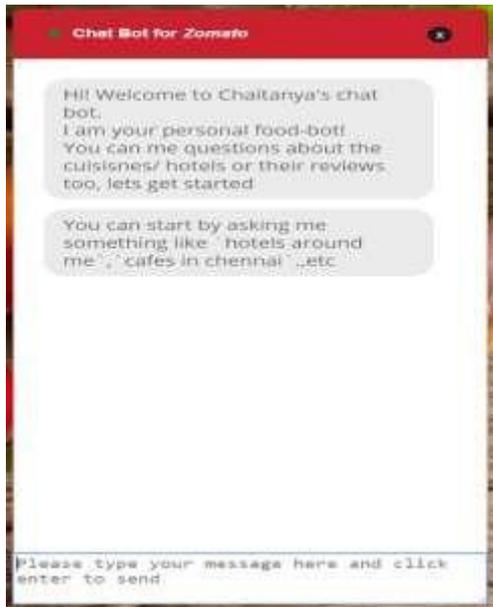


Fig 5 Welcome message from Bot



Fig 6 Bot responding for the query `Italian cuisines in Hyderabad` to user with 7 hotels

B. SUMMARY OF THE RESULT

For the test cases that are considered the images in the above section provides the screenshots of the results.

For the test cases that are considered the images in the above section provides the screenshots of the results.

a) Fig.5 showcases the welcome message from Bot when the webpage in the server is accessed. This helps in understanding that the bot is connected with the server and that the URL is working fine.

b) When the query `Italian cuisines in Hyderabad` is given by user user in Fig 6, from the results it is clear that the bot understood the context of the question and displays the

related information of list of 7 hotels with it's name, picture, cuisines in a card view and a read reviews button.

c) The output when the read reviews button is clicked ,from it an interpretation can be made that the bot understands the user question on which hotel reviews is mentioned and serves accordingly by displaying top 5 reviews for the same. When the user tries to make a query without giving the complete context of what the bot expects, the bot identifies the missing required informational parts and asks the user to enter it manually.

d. Finally when the image of the hotel is clicked the hotel's webpage is opened in the same page and has the details of the hotel.

VII. CONCLUSION

The model proposed in this paper concludes by acknowledging the importance and usefulness of chatbots in our day to day life. This provides us with a great number of opportunities where chatbots can be used very efficiently. The chatbots can be used in various domains starting from seeing the weather report to that of making transactions.

VIII. LIMITATIONS AND FUTURE SCOPE

The only limit to this is the imagination and the constraint is the complete wire flow. The chatbot started their development from simple string comparisons, and as of today huge neural nets and machine learning algorithms are used, but there's still a lot of scope for the development of these services and to make the conversation more human like.

This is possible by understanding the user interaction at a greater level and to understand their various slangs and dialects of the grammar. This makes more convenient and more comfortable for the user to interact with the bot. Also, training the bot in different languages must be done, as most of the times companies focus on developing a bot only in English makes it difficult to the user to convey their problems, so adding more languages and voice support can be a better way to grab more user's attention. This makes sense as a knowing a language on speaking else writing basis should never stand in the way of serving the user with a solution.

REFERENCES

1. Vijay Krishna Palepu, James A. Jones, Discriminating influences among instructions in a dynamic slice, Proceedings of the 29th ACM/IEEE international conference on Automated software engineering, September 15-19, 2014, Vasteras, Sweden
2. Muhammad Saleem, Rasheed Hussain, Vasir Ismail, Shaikh Mohsin, Cost-effective software engineering using program slicing techniques, Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, p.768-772, November 24-26, 2009, Seoul, Korea
3. László Vidács, Árpád Beszédés, Tibor Gyimóthy, combining preprocessor slicing with C/C++ language slicing, Science of Computer Programming, v.74 n.7, p.399-413, May 2009
4. Pierre Rousseau, A new approach for concurrent program slicing, Proceedings of the 26th IFIP WG 6.1 international conference on Formal Techniques for Networked and Distributed Systems, September 26-29, 2006, Paris, France

5. Kannan M. Moudgalya, Kiran L.N Eranki, Program Slicing Technique: A Novel Approach to Improve Programming Skills in Novice Learners, Proceedings of the 17th Annual Conference on Information Technology Education, September 28-October 01, 2016, Boston, Massachusetts, USA
6. Sandrine Blazy, Andre Maroneze, David Pichardie, Verified Validation of Program Slicing, Proceedings of the 2015 Conference on Certified Programs and Proofs, January 13-14, 2015, Mumbai, India real-time programs.
7. Christer Sandberg, AndreasErmedahl, Jan Gustafsson, Björn Lisper, Faster WCET flow analysis by program slicing, ACM SIGPLAN Notices, v.41 n.7, July 2006
8. Christopher Pietsch, Udo Kelter, Manuel Ohrndorf, Timo Kehrer, incrementally slicing editable submodels, Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, October 30-November 03, 2017, Urbana-Champaign, IL, USA
9. K. B. Gallagher, "Surgeon's assistant limits side effects", IEEE Software, vol. 7, pp. 64, May 1990.
10. M. Weiser, "Programmers use slicing when debugging", CACM, vol. 25, no. 7, pp. 446-452, July 1982.
11. K. B. Gallagher, J. R. Lyle, "A program decomposition scheme with applications to software modification and testing", Proc. 22nd Int. Conf. System Sciences, vol. II, pp. 479-485, 1989-Jan.
12. R. Grady, "Measuring and managing software maintenance", IEEE Software, vol. 4, Sept. 1987.
13. P. Hausler, "Denotational program slicing", Proc. 22nd Hawaii Int. Conf. System Sciences, vol. II, pp. 486-494, 1989-Jan.
14. S. Horwitz, J. Prins, T. Reps, "Integrating non-interfering versions of programs", Proc. SIGPLAN'88 Symp. Principles of Programming Languages, 1988-Jan.
15. S. Horwitz, J. Prins, T. Reps, "Integrating non-interfering versions of programs", ACM Trans. Programming Languages and Systems, vol. 11, no. 3, pp. 345-387, July 1989.
16. M. Weiser, "Program slicing", Proc. 5th Int. Conf. Software Eng., pp. 439-449, 1981-May.

AUTHORS PROFILE



Kannadasan R, P.hD, Vellore Institute of Technology
Assistant Professor (Senior)
E-mail id :kannadasan.r@vit.ac.in



Prabakaran N, P.hD, Vellore Institute of Technology
Assistant Professor (Senior)
E-mail id :prabakaran.n@vit.ac.in



Krishnamoorthy A, P.hD, Vellore Institute of Technology
Assistant Professor (Senior)
E-mail id :krishnamoorthy.arasu@vit.ac.in



Nerella Tarak Krishna, B.Tech, Vellore Institute of Technology, Mail Id: ntk05071998@gmail.com



Sai Vishnu Giri, B.Tech, Vellore Institute of Technology,
Mail Id: sai344@gmail.com



Konapalli Pavan Kumar Reddy, B.Tech, Vellore Institute of Technology, Mail Id: pkreddy430@gmail.com