

Chat Bot Using AWS

GOWTHAM S¹, HARISH R², ILAYA BHARATHI M³, BIJI ROSE⁴

^{1, 2, 3} Information Technology, Dr N. G. P Institute of Technology

⁴ Assistant Professor, Dr N.G.P Institute of Technology

Abstract- *This E-Booking with chat bot project is an Artificial Intelligence based web application designed to aid the car rental company to enable renting of cars. It has a user-friendly interface which helps the user to communicate with our bot and rent them for specific period. Communication can be done by chatting or by means of speaking. The ability of a bot to understand human language as it is spoken was done with N L P (Natural Language Processing). The categories of rental cars will be provided while speaking with our bot. Based on the type of car required by the customer, the user shall be able to make bookings. The bot shall also check for the availability of the car and rent the car to the customer. This bot shall ask the user about some important information such as date and time of journey, type of car etc. Also, it will need an identification number. Using these details, the tool shall help the customer to book a car for the journey. It can also keep the information of number of bookings in current month or in last 6 month or in last year. This helps them to track company business and their earning in any month or in any year. Based on this information they can take decision regarding their business development. It also provides another feature that the admin can manually chat with their customers in case of any technical difficulties. It also acts as a bridge for both customers and the company. It saves both time and labor. The main advantage is that the user shall be able to choose a car depending on his budget.*

I. INTRODUCTION

Artificial intelligence chat-bot is a technology that makes interactions between man and machines using natural language possible. From literature, we found out that in general, chat-bot are functions like a typical search engine. Although chat-bot just produced only one output instead of multiple outputs/results, the basic process flow is the same where each time an input is entered, the new search will be done. Nothing

related to previous output. This project mainly focuses on enabling chat-bot to become a search engine that can process the next search with the relation to the previous search output. In chat-bot context, this functionality will enhance the capability of chat-bot's input processing.

1.1 WHAT IS A chat-bot

A chat-bot is a computer program that simulates human conversation through voice commands or text chats or both. chat-bot, short for chatterbot, is an Artificial Intelligence (AI) feature that can be embedded and used through any major messaging applications. There are a number of synonyms for chat-bot, including "talkbot," "bot," "IM bot," "interactive agent" or "artificial conversation entity."

1.2 CHAT-BOT IN USE

A chat-bot is an automated program that interacts with customers like a human would and cost little to nothing to engage with. chat-bots attend to customers at all times of the day and week and are not limited by time or a physical location. This makes its implementation appealing to a lot of businesses that may not have the manpower or financial resources to keep employees working around the clock.

A chat-bot works in a couple of ways: set guidelines and Machine Learning. A chat-bot that functions with a set of guidelines in place is limited in its conversation. It can only respond to a set number of requests and vocabulary and is only as intelligent as its programming code. An example of a limited bot is an automated banking bot that asks the caller some questions to understand what the caller wants to be done. The bot would make a command like "Please tell me what I can do for you by saying account balances, account transfer, or bill payment." If the customer responds with "credit card balance," the bot would not understand the request and would proceed to either repeat the command or transfer the caller to a human assistant.

1.3 SUCCESS RATE OF CHAT-BOTS ON WEBSITES

The digitalization of all the industries has scattered its sweep establishing a rostrum where innovation has become the essence of development. Analogous to that is the rise of chat-bots, which is becoming a vibrant option to deal with customer service and is in the process of elaborating its functions to increase efficiency. It is profoundly definite from the recent study done by Gartner that chat-bots would be in charge of managing 85% of all the customer interactions by 2020. This evolution has happened about in various stages. The hatch of this technology initiated as early as 1950 with Alan Turing's intelligent machine which was then succeeded by another intelligent machine that could establish a quality interface between a machine and a human, making it more tangible. The graph did not show a steep until now, attributing its presence to the increasing demand for customer-oriented processes and competition.

1.4 THE UPSWING IN THE USE OF MESSAGING APPLICATIONS

The preferred channel of communication is what the messaging applications have transformed into. The number of people getting connected through these platforms is amplifying rapidly. chat-bots operate on an identical methodology which contains enormous prospective for market penetration.

1.5 AUGMENTING SERVICES OF CONVERSATIONAL INTERFACES

Facebook's annual developer conference has bolstered the need for technologies that whirl to establish more human-like dialogues and discourse. The superior level of chat-bots that is illustrated by 'Siri' and 'Cortana' are the versions that demonstrate the dialogue discourse in the voice form.

1.6 PROPELLING GROWTH OF ARTIFICIAL INTELLIGENCE

The statistics of AI market that blankets all the major technologies such as Machine learning, Image recognition, and speech processing are predicted to hit a mark of \$5.05 billion by 2020 as per the studies released by Statista.

1.7 CUSTOMER ORIENTED PROGRESSION

The call center approach of staying bounded with the consumers has been arched whilst a few years. The emphasis on customer experiences and rendering towards a differential hem when in strife to excel amongst the intense competition has led to the rise of chat-bots. Cost-effective, and user-friendly as prime characteristics chat-bots serve the end user as the first point of contact with the reduced hassle and a much-simplified process.

II. HARDWARE SPECIFICATION

The Hardware specification is made simple from basic mobile users to advance computer system users. The most basic hardware specification required for running this application is a mobile phone or a computer system with browser which support javascript.

The user can remotely use the web portal or the software in order to request about their car rental. The proposed system is implemented for the car rental communication in specified premises only. It can be extended to state level or national level. Other bookings such as hotel reservation also can be included in this project.

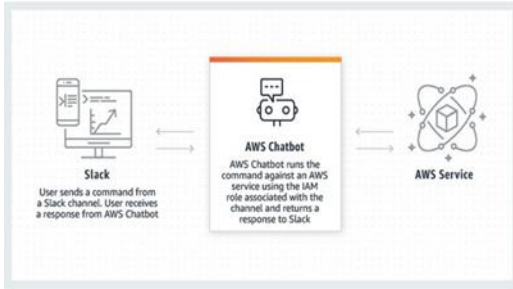
2.1 SOFTWARE SPECIFICATION

Operating System: Windows, Linux or Mac.
Server/DB: Amazon Lex.
Front End: CSS, JAVASCRIPT, JQuery.
Back End: Amazon cloud services.

2.2 SOFTWARE DESCRIPTION

2.3.1 AMAZON WEB SERVICES

Amazon Web Services (AWS), the cloud platform offered by Amazon.com Inc , has become a giant component of the e-commerce giant's business portfolio. Having grown steadily in the 30-percent range the past few quarters, AWS is a frontrunner to other cloud computing platforms such as competitor Microsoft Azure. AWS can be broken into three main products: EC2, Amazon's virtual machine service, Glacier, a low-cost cloud storage service, and S3, Amazon's storage system. AWS is so large and present in the computing world that it's far outpaced its competitors. As of February 2020, one independent analyst



reports AWS has over a third of the market at 32.4%, with Azure following behind at half that amount 17.6%, and Google Cloud at 6%.



2.3.2 AMAZON LEX

Amazon Lex is a service for building conversational interfaces into any application using voice and text. Amazon Lex’s high-quality speech recognition and natural language understanding capabilities make it possible to build powerful interfaces to applications. You can add a voice or text chat interface to create bots on mobile devices that can help customers with many basic tasks, such as accessing their bank account, booking tickets, ordering food, or calling a cab. Amazon Lex integrates with ‘Amazon Cognito’ so you can control user management, authentication, and sync across all devices. Here we are excited about utilizing evolving speech recognition and natural language processing technology to enhance the lives of customers. In our project Amazon Lex represents a great opportunity to deliver a better service to customers at the right time and in the right place through Chat Bot with the help of ‘Amazon Web Service’ Cloud System.

- Benefits of using Amazon Lex

1. Easy to use:

Amazon Lex provides an easy-to-use console to guide you through the process of creating your own chat-bot in minutes, building conversational interfaces into your applications. You supply just a few example phrases and Amazon Lex builds a complete natural language model through which your user can interact using voice and text, to ask questions, get answers, and complete sophisticated tasks.

2. Seamlessly deploy and scale:

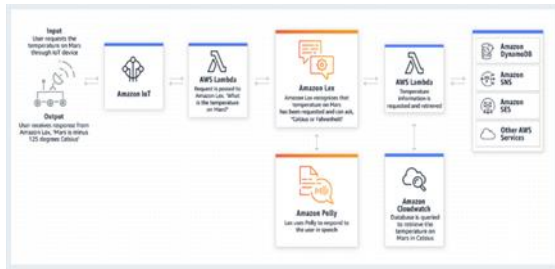
With Amazon Lex, you can build, test, and deploy your chat-bots directly from the Amazon Lex console. Amazon Lex enables you to easily publish your voice or text chat-bots to mobile devices, web apps, and chat services such as Facebook Messenger, Slack, and Twilio SMS. Once published, your Amazon Lex bot processes voice or text input in conversation with your end-users. Amazon Lex is a fully managed service so as your user engagement increases, you don’t need to worry about provisioning hardware and managing infrastructure to power your bot experience.

3. Built in integration with aws:

Amazon Lex provides built-in integration with AWS Lambda, AWS MobileHub and Amazon CloudWatch and you can easily integrate with many other services on the AWS platform including Amazon Cognito, and Amazon DynamoDB. You can take advantage of the power of the AWS platform for security, monitoring, user authentication, business logic, storage and mobile app development.

4. Cost effective:

With Amazon Lex, there are no upfront costs or minimum fees. You are only charged for the text or speech requests that are made. Amazon Lex pay-as-you-go pricing and low cost per request make it a cost effective way to build conversational interfaces anywhere. With the Amazon Lex free tier. You can easily try Amazon Lex without any investment.



2.3.3 NODE JS

As an asynchronous event-driven JavaScript runtime, Node.js is build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.

Node.js is similar in design to, and influenced by, systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems, there is always a blocking call to start the event-loop. Typically, behavior is defined through callbacks at the beginning of a script, and at the end a server is started through a blocking call like EventMachine::run(). In Node.js, there is no such start-the-event-loop call. Node.js simply enters the event loop after executing the input script. Node.js exits the event loop when there are no more callbacks to perform. This behavior is like browser JavaScript — the event loop is hidden from the user.

HTTP is a first-class citizen in Node.js, designed with streaming and low latency in mind. This makes Node.js well suited for the foundation of a web library or framework.

Node.js being designed without threads doesn't mean you can't take advantage of multiple cores in your environment. Child processes can be spawned by using our `child_process.fork()` API, and are designed to be easy to communicate with. Built upon that same interface is the cluster module, which allows you to share sockets between processes to enable load balancing over your cores.

III. FUNCTIONAL DESIGN

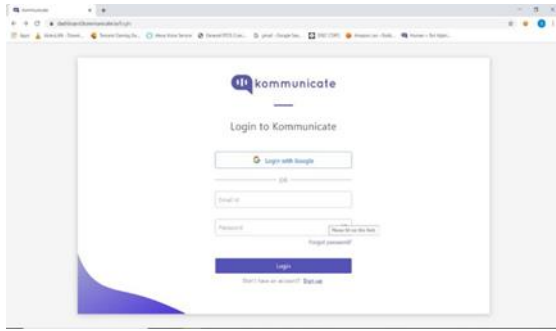
Functional Design is a paradigm used to simplify the design of hardware and software devices such as computer software and increasingly,3D models. A functional design assures that each modular part of a device has only one responsibilities and performs that responsibility with the minimum of side effects on other parts. Functionally designed modules tend to have low coupling.

The standard way to assure functional design is to review the description of a module. If the description includes conjunctions such as “and” or “or”, then the design has more than one responsibility and is therefore likely to have side effects. The responsibilities need to be divided into several modules in order to achieve a functional design.

IV. INTEGRATING BOT WITH OUR WEB APPLICATION

After configuring lambda function with our bot, we could integrate our bot in our concerned web application. We have created a website called Communicate.io . In the back end we have a done a coding part we could easily integrate our bot. It just normally asks Access Id(This will be provided by amazon while purchasing amazon cloud services), Secret access key(This will be provided while creating a bot in amazon lex), Region(This is a server where we developed our bot), Name of the bot in amazon lex, alias name. After completing this form we can directly integrate our bot profile. This is a step where our bot gets deployed in our web application.

4.1 LoginPage



V. CONCLUSION & FUTURE ENHANCEMENT

5.1 CONCLUSION

This study is a step further to improve the sustainability of car sharing as an environmental friendly transportation service, thus contributing to sustainability within our society.

Moreover, our study provides a valuable method for practitioners. They can generate more data to “stress-test” their system via simulation. This could be helpful to prepare for events (e.g., trade fairs), where a higher vehicle demand is anticipated.

5.2 FUTURE ENHANCEMENT

This work presents a research agenda based on interviews with experts in the field in order to guide future work from an industrial viewpoint, as current agenda-setting contributions fail to include this perspective. Also, we derived guidelines based on expert interviews to provide insights for companies seeking to implement chat-bots in customer care.

To explore, our expert panel and literature review showed us that there is a broad consensus that intelligent chat-bots bare great potential for creating value for companies in customer care. While we see the existing research as highly important, our literature review also indicates that the industrial perspective should be more strongly included in identifying needed research. In this way, practice-based requirements can provide insights that may not have been captured in scientific literature, and thus advance the IS research discipline.

REFERENCES

- [1] G.Daniel, J. Cabot, L. Deruelle, and M. Derras, “Multi-platform chat-bot modeling and deployment with the jarvis framework,” in *Advanced Information Systems Engineering (Lecture Notes in Computer Science)*, vol. 11483, P. Giorgini and B. Weber, Eds. Rome, Italy: Springer, Jun. 2019, pp. 177–193, doi: 10.1007/978-3-030-21290-2_12.
- [2] O. Diaz and F. M. Villoria, “Generating blogs out of product catalogues: An MDE approach,” *J. Syst. Softw.*, vol. 83, no. 10, pp. 1970–1982, Oct. 2010, doi: 10.1016/j.jss.2010.05.075.
- [3] D. Falessi, N. Juristo, C. Wohlin, B. Turhan, J. Münch, A. Jedlitschka, and M. Oivo, “Empirical software engineering experts on the use of students and professionals in experiments,” *Empir Softw. Eng.*, vol. 23, no. 1, pp. 452–489, Feb. 2018, doi: 10.1007/s10664-017-9523-3.
- [4] J.Hutchinson, J. Whittle, and M. Rouncefield, “Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure,” *Sci. Comput. Program.*, vol. 89, pp. 144–161, Sep. 2014.
- [5] P.Jackson and I. Moulinier, *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*, vol. 5. Amsterdam, The Netherlands: John Benjamins, 2007.
- [6] D.Kavaler, S. Sirovica, V. Hellendoorn, R. Aranovich, and V. Filkov, “Perceived language complexity in GitHub issue discussions and their effect on issue resolution,” in *Proc. 32nd IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Oct. 2017, pp.
- [7] A.Kleppe, *Software Language Engineering: Creating Domain-Specific Language Using Metamodels*. London, U.K.: Pearson, 2008.
- [8] L. C. Klopfenstein, S. Delpriori, S. Malatini, and A. Bogliolo, “The rise of bots: A survey of conversational interfaces, patterns, and paradigms,” in *Proc. Conf. Designing Interact. Syst. (DIS)*, 2017, pp. 555–565.