

IMPLEMENTATION OF CHATBOT USING AWS AND GUPSHUP API

Pramod K.¹, Akash Hegde², Sandhya S.³, Dr. Shobha G.⁴

¹⁻⁴Department of Computer Science and Engineering, R. V. College of Engineering, Bengaluru, India

ABSTRACT: A chatbot can be defined as a program developed to carry out conversations with a human using either audio or text. There exist numerous chatbots which are used for various purposes such as e-commerce, customer support, design, communication, finance, education, analytics, and so on. Furthermore, many companies use chatbots for their internal operations, for human resources, for customer support and more recently, support for Internet-of-Things (IoT) operations has also been added. Bearing in mind the existing chatbot applications with respect to productivity, the aim is to develop a chatbot for various operations related to productivity and project analysis within an organization, such that it can be integrated with CA Technologies Rally (Agile Central). It can be used for checking tasks and defects, generating reports and obtaining notifications. In the proposed work, the chatbot is built using Gupshup Bot Builder API which deploys it on to Amazon Web Services (AWS) Cloud, and then, it is integrated with Rally. Natural language processing (NLP) is used by the chatbot in general command interactions with the user, thereby eliminating the need for a fixed database of interaction commands.

KEYWORDS: *chatbot; cloud computing; natural language processing; project analysis; project management.*

I. INTRODUCTION

A chatbot is known by many names in the current world. It can be called as a smartbot, chatterbot, talkbot, interactive agent, conversation agent, conversational interface, artificial conversational entity, or simply a bot. A chatbot can be described as a developed program or a human-created artificial intelligence (AI) that uses various technologies to mimic a conversation that a human would have with another human. It can carry out the conversation either by audio or by text.

Chatbots are designed and developed such that they can simulate the way in which a human behaves in a regular conversation. This allows the chatbots to pass the Turing test, such that they are indistinguishable from a human. Chatbots are commonly used in dialog systems, such as customer service or information acquisition, which are two of the most practical applications in today's world. Recently, more chatbots are being developed that use complex NLP systems for their processing, as compared to traditional chatbot systems, which scan for keywords when the input is provided, then check for the reply which contains the most matching keywords, or a pattern of words, from an existing database on a server.

The term chatterbot was first introduced to the world by Michael Mauldin in 1994; he used this term to describe conversational programs. He is the creator of the first Verbot known as "Julia." The first chatbot was "ELIZA," developed by Joseph Weizenbaum in 1966.

A. Accessibility and Usage

Virtual assistants are usually used to access chatbots; they can also be accessed using messaging apps or messengers such as Facebook Messenger, or also using apps and internal websites of individual organizations.

Chatbots are generally classified into many categories based on their usage. Conversational commerce, also known as "e-commerce via chat" is one of the major categories. It can also be observed that they are used for communication purposes, for financial purposes and design purposes. Some people use chatbots for their personal tasks. They can also be used in education sector, marketing campaigns, sports events organization, entertainment industry, data analytics, multiplayer

games where text-based chat is needed, customer support over the internet, health care, Human Resources (HR) and management, news discussions, productivity analysis, developer tools, social media networking, travelling navigation and many other utilities. Some chatbots are used for Business-to-Consumer (B2C) customer service, marketing and also, sales of business products. The chatbots of many companies run on messaging apps such as Messenger for Facebook, WhatsApp Messenger, Slack app, WeChat messenger, Telegram, LiveChat, and Line messenger. The bots are generally present in the contacts of the user in the app, but they can also participate in a group chat, catering to a large number of people. They have become increasingly popular these days. Banks and insurance agents, media companies, e-commerce websites, airline companies, hotel management and restaurants, shop retailers and owners, government agencies and health care providers use chatbots on a daily basis to do limited tasks such as answering simple questions raised by the customers, increasing customer services engagement, promotional purposes, and also, by offering additional ways with which the users can order from them.

Furthermore, many companies use chatbots for their internal operations, for human resources, for customer support and more recently, support for Internet-of-Things (IoT) operations has also been added. For example, Overstock.com, which specializes in e-commerce, has launched a chatbot recently, which is named "Mila"; it is primarily used to automate simple processes which are extremely time-consuming, for example, when an employee posts a request for a sick leave.

Fig. 1. shows different chatbots available in different domains. These chatbots provide services to the users based on the domain to which they are applied. The user would be able to provide voice commands to the bot available on the device, which interacts with the applications and provides service back to the user.

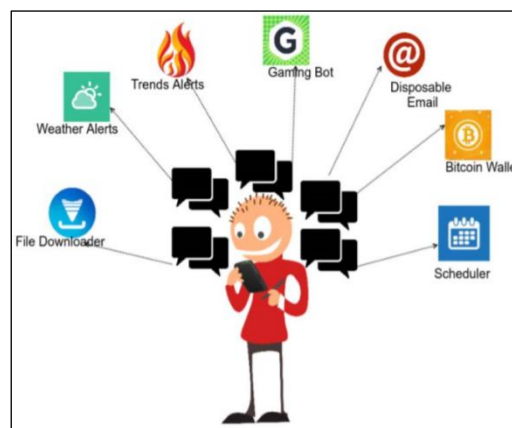


Fig.1. Chatbots in different domains

B. Organization of the Paper

This paper focuses on implementation of a chatbot using AWS cloud and Gupshup API. This chatbot is then integrated with CA Technologies Rally to be used for productivity and project analysis. Section II describes the related work that has taken place so far in terms of chatbots. Section III gives the methodology used to build the chatbot and the approach employed to integrate it with CA Technologies Rally. Section IV describes the implementation of the chatbot using the Gupshup Bot Builder API and CA Technologies Rally. Section V describes the results obtained from this implementation and is followed by Section VI, which concludes by giving the overview of the paper.

II. RELATED WORK

The topic of chatbots has drawn the attention of many researchers and thinkers. There are a number of review papers that describe the increasing use of chatbots in modern times. It has drawn many researchers to develop newer and better technology to incorporate chatbots to do menial tasks, provide customer care, give mental and emotional support, among others. There has also been an increased demand for Internet-of-Things (IoT) in recent days, and this has further alleviated the usage of chatbots for this purpose.

A novel proposal to develop a chatbot that takes in to account the context of how a conversation occurs is discussed. TensorFlow library with neural network [1] is used for developing the model and Natural Language Processing (NLP) techniques are used to determine the context, before giving a response. The chatbot thus developed can be used in small industries, it can also be used for automating customer care in businesses, wherein the chatbot handles user queries and thus, reduces the need of human labour and extra investment and expenditure.

A novel system architecture that tries to overcome the problem of solving grievances of a huge volume of users of any particular social media platform is proposed. This method analyses the messages [2] of each jabberd user to check for its actionability, where jabberd is a XMPP application server that is written in Erlang. If the messages are actionable, then the chatbot starts the conversation automatically with that particular user and helps the user to resolve any issues, by using Language Understanding Intelligent Service (LUIS) to interact with the user like a human. Their proposed system is implemented on AWS cloud publicly. This is done to provide an extremely robust, highly scalable and architecture that is extensible.

A review paper that discusses the overview of cloud-based chatbot technologies, programming of chatbots and challenges of programming in the current and future generations of chatbots is studied. The working of a chatbot using machine learning [3] in the Python programming language is described. Information about the challenges faced by the chatbots and their working is given, mainly the proper implementation and maintenance of natural language processing and machine learning techniques in chatbots. It is further stated that handling complex queries needs a lot of attention in the current scenario of chatbot development and also recommend the use of sentiment analysis as much as possible, in upcoming chatbots.

The past research done on chatbots or conversational agents using quantitative bibliometric analysis [4] is presented. The goal is to help future researchers identify the gaps for future development and research in the field of chatbots. The findings are presented and it is stated that there exists a potential research opportunity in the field of deep learning chatbots. The paper also gives several recommendations for future research based on the results.

A paper is presented that provides a review study of integrated applications which use chatbots that run on Artificial Intelligence Markup Language (AIML) [5]. These applications are prevalent in various fields such as cultural programmes and national heritage, e-learning courses, online functioning of the government, web-based and dialog models, framework for semantic analysis, framework for interaction, management of networks and modular architecture that can be adapted to any other architecture easily. The chatbots provide services, as well as interact with the users frequently and propose and determine solutions to problems through AIML-based intelligence. It is also stated that it is extremely popular with entrepreneurs these days, which employ these chatbots to help grow their business.

A comparison between Google Analytics and Adobe Analytics is made. An AIML-driven chatbot [6] is proposed that takes in raw data of the analytics as the input and enables bot users to receive business insights as the output by querying the bot.

A chatbot is used to control functions of electrical appliances over the Internet, employing IoT techniques [7]. The messages sent to the bot are processed using NLP techniques. The application is also embedded with security features that only allows access to the authorized users and also informs the users whenever an intruder is detected using motion sensors within the house.

A chatbot is built based on Recurrent Neural Network (RNN) principles [8]. Sequence-to-sequence Long Short-Term Memory (LSTM) cell neural network is used on Google's neural net word2vec. It is stated that model training times and its language model quality that is used for training affects the prediction output quality of the chatbot.

A. Existing Chatbots in India

This section describes the famous chatbot applications that currently exist in India. These can be categorized into two, based on the services or assistance provided to the public.

1. General Services

These types of chatbots are used to provide general services to the public, such as information about offers and coupons, interactive chat with the users and electronic bill payments. They usually consist of multiple services that help the public with many day-to-day activities. Some of the chatbots that fall under this category are as follows.

- "Sonia" provides information about coupons and offers from top online shops. It is deployed by developer Jiss Jose.
- "TOI Personal Assistant" chatbot represents Times of India. It is deployed by Haptik and integrated into the TOI mobile app. Apart from providing news, it also allows users to book cab services such as Ola and Uber, pay monthly bills such as TV and electricity. It accepts payment using either credit or debit card, net banking and e-wallets.
- "Ziman" is a chatbot representing Zicom. It is deployed by Haptik. The main aim of Ziman is to mimic a human conversation for people who do not feel safe and comfortable when travelling. It also aims at being available all the time to the users.
- "Rembo" is a chatbot deployed by Haptik, used to set reminders and task to-do's. It also shares jokes and motivational quotes.
- "Fun Bot" is a chatbot representing Maruti TechLabs. It is deployed by Maruti TechLabs. It asks some questions to understand the user's personality. Based on the answers provided, it recommends holiday destinations.
- "Brev" is a chatbot representing Brevity. It is deployed by Artificial Industry. It provides life hacks, how-to's, latest stories and information in categories like health, food, work, relationships, and technology.
- "Haptik Assistant" is a chatbot representing Haptik. It is deployed by Haptik. It is an assistant used to do many things such as setting reminders, booking cab services, trains, airlines, paying bills and finding nearby locations of interest.
- "Amy" is a chatbot used to talk to freely and ask for information like stock prices. It is deployed by Abhishek Bhattacharya.

2. Specific Assistance

These type of chatbots are used only for certain specific applications and/or services. The interaction between the user and the chatbot is usually limited to only the particular topic with which it is associated. Some of the chatbots that fall under this category are as follows.

- "Pathology Lab Chatbot" represents Dr. Lal PathLabs. It is deployed by Haptik on the Dr. Lal PathLabs website. Any visitor can login and check the status of his/her reports. It also guides the user through every step, thus enabling quicker resolution of queries. It is also helpful in finding nearby medical centres, for booking a check-up or a test or just browsing the catalog for information about the tests, check-ups and their respective prices.
- "Akancha" chatbot is used to represent Akancha Against Harassment. It is deployed by Haptik. It serves users who are seeking help, by using a chat-based interface that is personalised to their needs, and is extremely approachable. The bot can answer various queries regarding cyber security and safety, methods to contact the police, outlining of each person's rights, laws provided by the Indian Constitution. This bot is trained to accept requests

for help and has been designed in such a way that it mimics human behaviour with a personality that is comforting and empathetic towards the users.

- "Home Services Bot" is a chatbot representing Housejoy. It is deployed by Haptik. It provides booking for online home services like personal care, home needs and appliances.
- "Coke India" is a chatbot representing Coca Cola. It is deployed by Haptik. It allows users to ask questions about Coca Cola's quality standards, community campaigns, water usage and helps raise awareness among consumers.
- "Ask Me" is a chatbot representing Citibank India. It is deployed by Creative Virtual on the Customer Service Center of Citibank's website. It is designed to provide information to customer queries about Citibank products and services.
- "Rickfare" is a chatbot representing Rickfare Inc. It is deployed by Rickfare Inc. It calculates autorickshaw and taxi fares for Delhi, Pune, Mumbai and Ahmedabad.

III. PROPOSED METHODOLOGY

This section describes the methodology used in this paper to build the chatbot. It also gives some insights into the concepts used and the use cases that are implemented.

A. Objectives

This paper aims to develop a chatbot that can be used to implement work flow monitoring and project analytics, which can then be easily integrated with CA Technologies Rally (Agile Central). Proper authentication procedures have to be provided so that it is secure and does not leak information to the outside world. Deployment has to be done in a secure environment and only the authenticated users should be able to interact with the chatbot, or change its code or inherent functionalities and features. Three use cases are specified, which are given as follows.

- Use Case 1 – Each chatbot is usually assigned a particular "room," which is a pre-defined space in which the chatbot can interact with the users. A user should be able to add the chatbot to the room using the ID or name. The user should be able to interact with the chatbot using a set of commands as defined by the developer. The chatbot is expected to be utmost available and any outage should alert all the users about the unavailability. The chatbot is also expected to generate and preserve logs of the places it has been integrated onto.
- Use Case 2 – All users in a particular room should be able to interact with Rally. The actions of the users that are done on a day-to-day basis for internal operations, such as update tasks, discussions, defect/story status, are to be made available to all the users using the chatbot. Update tasks have to be reflected in Rally and the chatbot should confirm this updation. The chatbot should also be able to show the status of a particular field when requested for. If there is an invalid syntax used when chatting with the bot, an alert must be generated and a help file with the correct syntax must be shown to the users.
- Use Case 3 – A project manager must be able to generate various metrics such as iteration burndown, bugs for a developer in an iteration, developer points delivered, among others. This use case gets various metrics that will give insights into various metrics broadly based on productivity, efficiency and quality.

B. Chatbot Design

A chatbot is created similar to how a mobile application or a webpage is created. There are four stages in this process - designing the bot, building the bot, analyzing the bot and maintaining the bot.

- Designing the bot – The design process of a chatbot is done based on how a user interacts with the chatbot. The designer adds features to the chatbot based on the requirements specified and also the responses that have to be generated to the queries are outlined. Conversational design can be considered as a superset of this process. Chatbot designers use several design tools and software, which provide previews of how the chatbot is going to function. This helps in collaboration of the team that is designing the chatbot, thus speeding

up the design process. User testing is also a crucial aspect in chatbot design. It can be carried out in a similar fashion as that of user testing of a GUI, for example.

- Building the bot – In the chatbot building process, two tasks take place: the first task is to understand how the user is able to interact with the chatbot with some queries and the second task is to produce the right answer to the respective queries. In order to accomplish the first task, an NLP Engine is usually used, which uses NLP to understand how to adapt to different user queries. In order to accomplish the second task, the chatbot is programmed to generate different types of responses to various queries posed by the users. This allows the chatbot to select which response has to be provided whenever a query is input to the chatbot.
- Analyzing the bot – In the analysis stage, the usage of the chatbot is continuously checked for any existing problems or flaws. This is done in order to reduce the number of bugs that the users might experience during their interactions with the chatbot. Thus, the user experience is enhanced due to this process, which will aid in more number of users employing the chatbot for their tasks.
- Maintaining the bot – The requirements, services and products of any company keeps changing, due to the fast-paced scenario of the world. This means that the chatbots that are employed by these companies will have to be updated regularly to be able to provide uninterrupted service. The traditional platforms used to develop chatbots need constant maintenance. Either a service provider can take care of this or a chatbot training team within the organization can help in this maintenance. These procedures utilise a lot of company resources, incurring an increased cost to the company. In order to avoid these costs, small start-up companies now use artificial intelligence for maintenance, wherein the AI develops chatbots which are self-learning.

C. Natural Language Processing in Chatbots

NLP technology is basically used to empower chatbots. When a user inputs some text to the chatbots, they process the text in the form of parsing it, which is followed by a response generated after a complex series of algorithms which interpret the user input and determine the most appropriate inference as to what the user means or wants. Then, the chatbot determines an appropriate response which can be given back to the user. Some chatbots are extremely similar to a human in terms of their authentic conversation, which makes it difficult to determine if the agent that is on the other side of the conversation is a chatbot or a human.

Inherent chatbot technology is distinctly different from NLP technology, but this also means that the chatbot technology can improve as quickly as the NLP technology; if there are no continuous developments in NLP, chatbots will not be able to understand the nuances in spoken and written dialogue.

This is where many NLP-based applications perform poorly. Any application or system that depends upon the ability of a machine to process speech is fairly likely to falter when dealing with metaphors and similes in the elements of speech. Even though these limitations exist, chatbots are becoming complex, more responsive to commands and acting natural.

Fig.2. depicts the general working of chatbots. The user input provided at the front-end is understood by the NLP layer, which then checks for the existing data in the knowledge base or the Central Management Server (CMS). If the chatbot has already interacted with the user earlier, the history and analytics of the conversation would be stored in the data store.

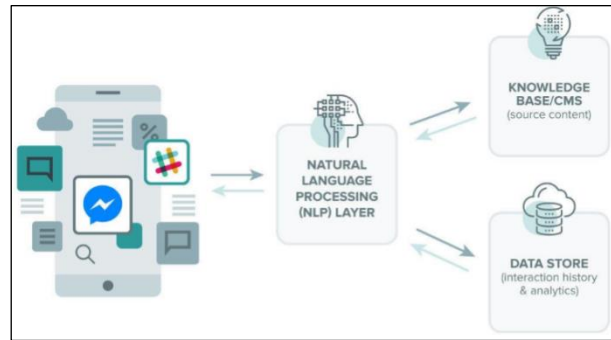


Fig.2. Working of chatbots

D. Methodology Involved in Creating the Chatbot

This subsection describes the main methodology involved in the establishment of a working chatbot that has been implemented in this paper.

Gupshup API is used to build the chatbot, which internally uses git and there is no presence of a package manager. Thus, the deployment process has to be done manually, as described in Section IV.

Once an account is created in Gupshup.io, a secure and unique API key is provided to the user in order to access the data necessary to deploy the chatbot. The access is only limited to the user's account and a git repository is already created in the back-end.

The deployment is done on AWS EC2 instances, where the replies from the chatbot are received as messages in AWS. These messages can be verified in the console logs that are generated whenever there is any interaction with the chatbot. There are multiple access points to this particular instance, which can be accessed via different messenger platforms. The chatbot will respond to that particular platform accordingly. There are a limited number of requests that can be sent. Rally offers a time-limited trial for this particular reason. It is also worth noting that AWS internally uses a read-only file system, thereby keeping the data and messages secure and not allowing the user to tamper with the existing files and repositories.

Node.js is mainly used because it comes with a default package manager and is easy to incorporate both server-side and client-side functionalities for the chatbot.

E. Importance of Chatbots

Chatbot applications are very important as they enhance customer experience by providing streamlined interactions between people and services. They also offer opportunities to companies to enhance customer service by engaging them with an automated bot that improves and aids their operations; this also results in reduced cost to the company. When a chatbot solution performs both of these tasks, it can be said that it is successfully operational. Human support is also crucial and essential because of their necessary intervention in building, configuring, constant training the chatbot to optimise and improve the system.

IV. IMPLEMENTATION

This section describes the tools used for implementation and also describes the process of building the chatbot and its integration with Rally.

A. About Gupshup Bot Builder

Gupshup.io is a web platform that offers a tool, which is known as the Bot Builder; this is primarily used to create and configure bots. This tool includes a code editor to write and edit code, a mechanism to publish the bot, analyse and diagnose the built bot employing the available tools, thus simplifying the overall process of building a chatbot.

1. Features of the Bot Builder

A usual process when a bot has to be built is to set up a developer environment, find and install necessary packages and libraries, set up space for the server, among many others. Gupshup reduces the burden on developers by taking care of all the above mentioned time-consuming processes. The IDE of the bot builder has a few features that make this possible. The pre-installed libraries such as a JavaScript async library and node-wit that is packaged with the code is very helpful.

The builder also provides a terminal to install new npm packages, to use git commands, to maintain the codebase and access bot logs for debugging and logging. The creation of a new bot comes with template code for common processes, which include helper methods. The IDE Bot Builder also provides single-click secure server deployment for the chatbot, thereby making it clear that the user need not set up his/her server.

The IDE Bot Builder is built on top of Amazon AWS Lambda and provides automated hosting. Gupshup Proxy Bot is used to test the chatbot, after deployment. The conversational aspects of the chatbot can be tested using the built-in chat widget given in the IDE Bot Builder.

B. About CA Technologies Rally

Agile methodology uses a concept called sprints – these are small, iterative periods of time, wherein the focus is on achieving small objectives such that the bigger and major objective is reached through these iterations. CA Technologies' Rally is an agile project management platform which aims at using this concept of agile, by delivering an early version of the product and then performing improvements on it.

CA Technologies Rally is now called CA Agile Central. The main idea of this software is to prioritize the tasks and complete these tasks following a plan given by the management of the company. Rally gives the option to a developer to set up a hierarchy for his/her project and portfolio. An overall roadmap can be set using Rally and continual monitoring of how well the teams are doing can be carried out.

Tracking of the releases and metrics are also present in Rally, which output an extensive report on user stories, tasks, defects, progress, dependencies, alignment and overall progress of a project.

C. Stages of Deployment

This paper aims at creating a chatbot used for project and performance analysis. It will also strive towards integrating it with the agile project management solution Rally to enable ease of use, single end point and enhanced analytics. There are mainly three stages in the development – building the bot, secure integration with Rally (CA Agile Central) and deploying it on AWS via Gupshup.

1. Building the Chatbot

The chatbot is built using the bot builder platform Gupshup.io. An index file is created to navigate to the respective modules. The bot is built to be efficient at taking input and providing output to the end user. Coding is done in Node.js to facilitate easier debugging for anyone with knowledge of Java and JavaScript concepts.

The general interactions with the bot are defined in a script that has a bit of NLP involved in it, to provide accurate answers to the end user's input. The application-specific interactions require the use of syntax to interact with the bot, which is specified in the documentation manual.

2. Secure Integration with Rally

The chatbot is then securely integrated with Rally (CA Agile Central) in order to facilitate easier access to projects, user stories, tasks and defects. Rally provides an agile project management solution that can be effectively used to make the work faster and more productive. An end user can chat with the bot and check the status of on-going projects, user stories, tasks and defects. Alternatively, the user can login to Rally to verify the same.

3. Deployment on AWS via Gupshup

Finally, the chatbot is deployed on AWS via Gupshup platform and can be accessed by any authenticated user to perform the specified tasks. It can also be used for testing purposes and raising defects, in order to be rectified by the project managers.

D. Authentication Procedures

Gupshup provides an API key with which an authenticated user can access the code built to create the chatbot. Once the code is completed, it can be deployed on AWS via Gupshup. Only the authenticated users must be able to access the project analytics information, metrics and charts. This is taken care of by AWS, wherein the user must input credentials in order to gain access to the chatbot services offered by the developers.

E. Interactions with the Chatbot

The chatbot is functional to a specific domain and cannot be interacted outside this domain. There are basically two types of interactions that can be done with the chatbot – general interactions and command-specific interactions.

1. General Interactions

These interactions with the chatbot need not have to follow a specific syntax. The queries for these interactions are usually about general topics and the chatbot responds to these queries without any need of technical information. For example, if "Hi!" or "Can you help me?" is sent to the chatbot by the user, the chatbot may respond with "Hi!" or "Yes please!" respectively.

Fig.3. shows an example of a general interaction between the chatbot and the user. It can be observed from the figure that the chatbot learns to respond with "Hi" even when two different inputs of the greeting are given to the chatbot.

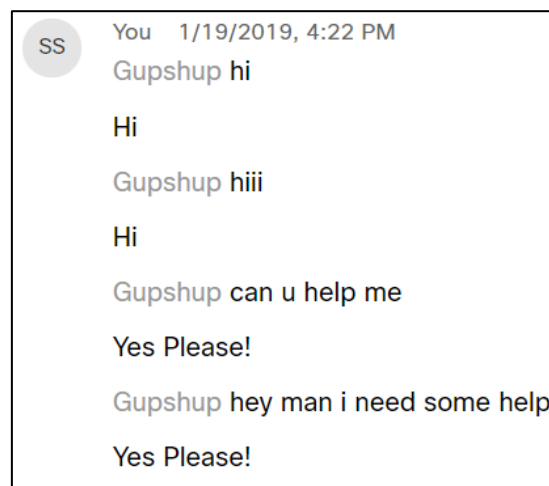


Fig.3. Example of a general interaction

2. Command-specific Interactions

These interactions with the chatbot are with respect to the project management aspect. The bot is programmed to respond to commands regarding creation, updation and deletion of projects, user stories, tasks and defects. For example, "create story Workspace". Fig.4. shows an example of a command-specific interaction. Here, it can be observed that the list of user stories is queried to the chatbot, which responds with a list of existing user stories. The status of the story "newworld" is then queried to the chatbot and it provides the necessary information as the output. Invalid queries to the chatbot are ignored and an alert message with a link to the help manual is generated.

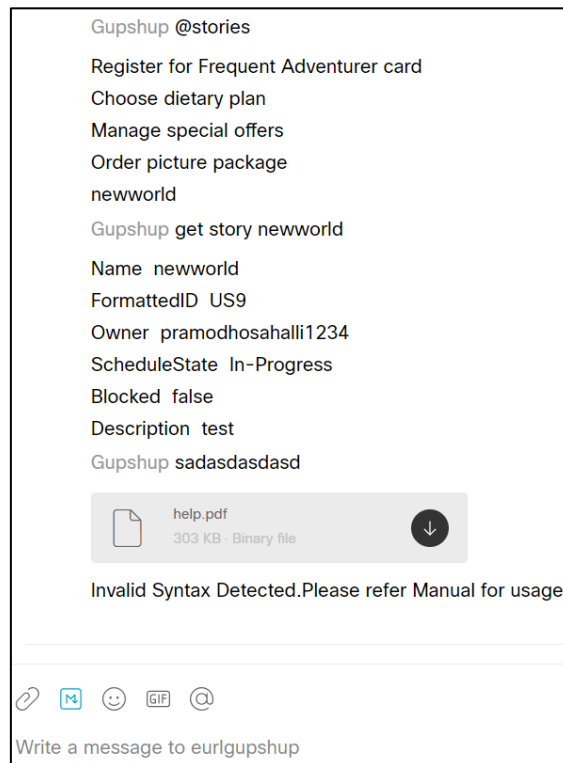


Fig.4. Example of a command-specific interaction

V. RESULTS

This section describes the results of deployment of the chatbot on AWS via Gupshup platform and its secure integration with CA Technologies Rally.

A. Deployment on AWS via Gupshup

The chatbot has been built using Gupshup Bot Builder platform, primarily using Node.js. It has been deployed on AWS via the Gupshup platform and is successfully able to interact with the user for general interactions and command-based interactions. The chatbot is functional within a specific domain and responds to messages given by the user.

B. Secure Integration with Rally

The chatbot has been successfully and securely integrated with CA Technologies Rally (Agile Central) platform which allows an authenticated user to perform the following commands - create, use, update, check for status, and delete. These commands are applicable to user stories, defects and tasks.

Furthermore, reports can be generated to check the output of the command-specific interactions. These are generated as a HTML file and can be downloaded to the local disk in the form of a JPEG or PNG image, XL sheet or CSV, among many other formats. *Fig.5.* and *Fig.6.* depict this scenario. Here, the report is generated for the defects that are present in a particular iteration. It is then visualized as a pie graph, which shows that the open defects are visualized with an orange colour, whereas the fixed defects are visualized with a green colour. *Fig.7.* also represents an example of a generated report, but with the presence of three stages of defects.

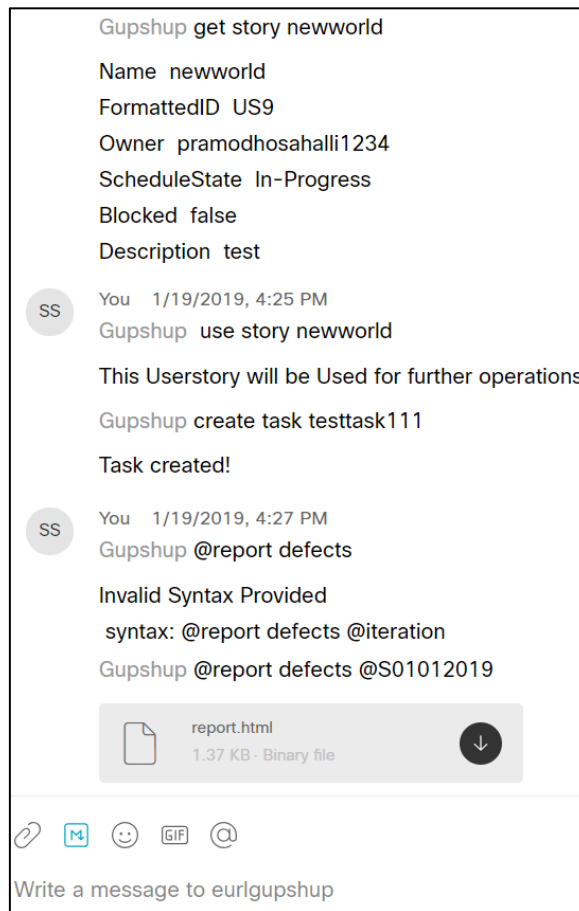


Fig.5. Command to generate report

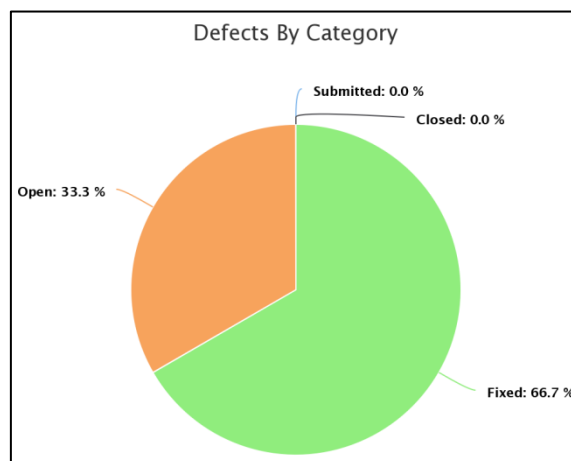


Fig.6. Example of a generated report

In the case of multiple users working on the same user story or task, the administrator account always receives notifications which specify the activity that is taking place in the system.

Fig.8. depicts the notifications. It can be observed here that whenever any changes have been made to a particular user story or a task, the chatbot immediately provides the notifications.

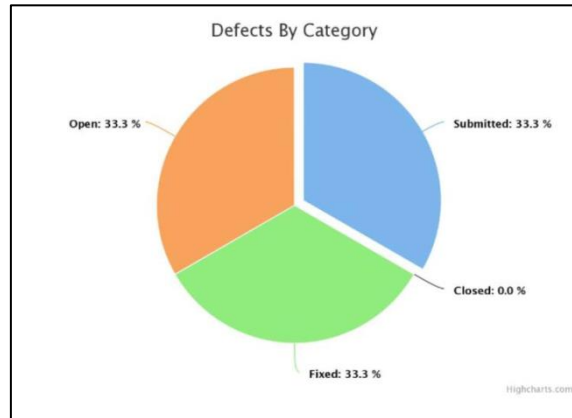


Fig.7. Another example of a generated report

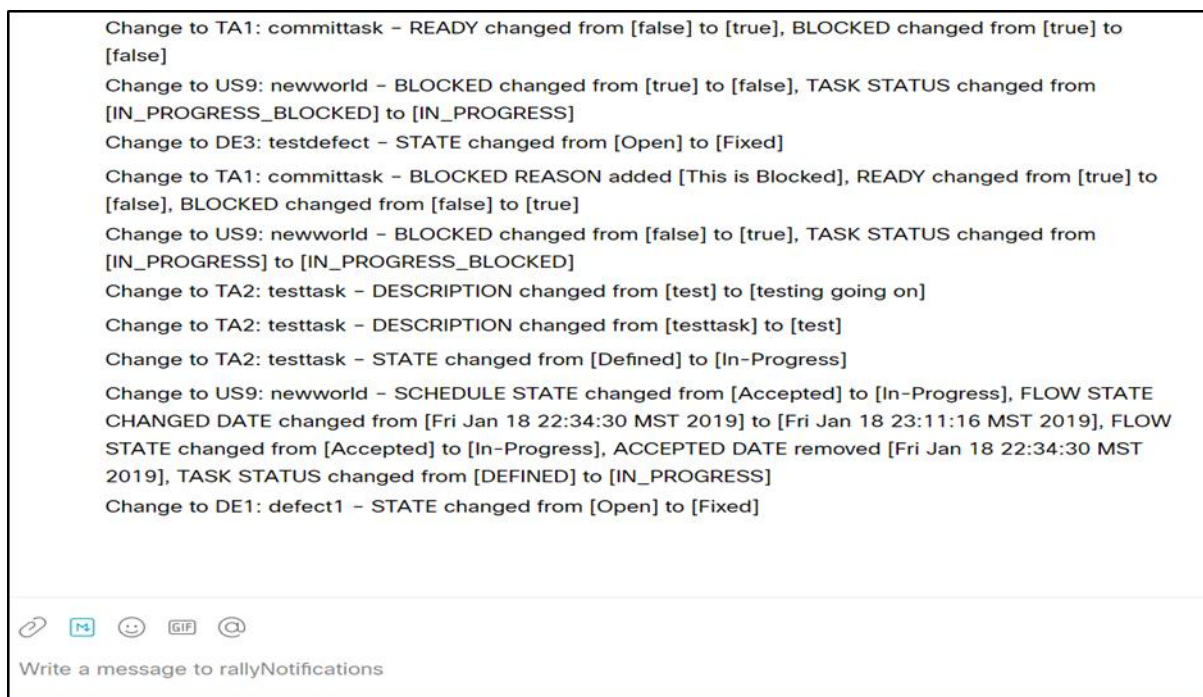


Fig.8. Example of notifications

C. Analysis

The chatbot thus created has been observed to respond to user's commands within 2-3 seconds, depending on the strength of the internet connection. The deployment on AWS via Gupshup platform allows any authenticated user to securely interact with the bot and enter the set of commands. In the event of a wrong entry by a user, the chatbot provides the documentation manual which gives the correct syntax and example usage of the particular command.

Additionally, the chatbot provides reports of metrics as and when needed. This is helpful for productivity analysis. Notifications are important to the administrator to manage the projects that are going on simultaneously in the organization. They can also provide details regarding the changes that are made to the projects in real time. This helps in project analytics, productivity and management.

VI. CONCLUSION

Chatbots are conversational agents which allow humans to interact with them and obtain some useful information or perform some specific tasks or services. The advent of popular technologies such as natural language processing and deep learning has seen the incredible rise of the usage of chatbots in many applications over different domains.

A chatbot has been developed with respect to the requirements for productivity and project analytics. Node.js is primarily used to develop the code for the chatbot, which has been integrated with CA Technologies Rally and deployed on AWS via Gupshup platform. The code is kept secure via authentication procedures on Gupshup, as well as AWS. The deployment is made on AWS only after using the API key provided inherently by the Gupshup platform, which has capability of automated hosting to AWS.

The chatbot is capable of interacting casually as well as pertaining to the productivity aspects like creation and updation of user stories, tasks and defects, among other functionalities. Notification alerts are sent to the user whenever there is any update or if the user inputs an invalid command. A documentation manual is also provided, in case the user is unable to recollect the usage syntax of the commands. Future scope of this chatbot would be to include more features to help in the productivity analysis within the organization.

REFERENCES

1. R. Singh, M. Paste, N. Shinde, H. Patel and N. Mishra, "Chatbot using TensorFlow for small Businesses," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2018, pp. 1614-1619.
2. G. M. D'silva, S. Thakare, S. More and J. Kuriakose, "Real world smart chatbot for customer care using a software as a service (SaaS) architecture," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, 2017, pp. 658-664.
3. A. M. Rahman, A. A. Mamun and A. Islam, "Programming challenges of chatbot: Current and future prospective," 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 2017, pp. 75-78.
4. H. N. Io and C. B. Lee, "Chatbots and conversational agents: A bibliometric analysis," 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 2017, pp. 215-219.
5. M. S. Satu, M. H. Parvez and Shamim-AI-Mamun, "Review of integrated applications with AIML based chatbot," 2015 International Conference on Computer and Information Engineering (ICCIE), Rajshahi, 2015, pp. 87-90
6. R. Ravi, "Intelligent Chatbot for Easy Web-Analytics Insights," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, 2018, pp. 2193-2195.
7. C. J. Baby, F. A. Khan and J. N. Swathi, "Home automation using IoT and a chatbot using natural language processing," 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, 2017, pp. 1-6.
8. M. T. Mutiwokuziva, M. W. Chanda, P. Kadebu, A. Mukwazvure and T. T. Gotora, "A neural-network based chat bot," 2017 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, 2017, pp. 212-217.