**Project Report**

**Introduction:**

**Overview:**

Companies respond to the queries of their customers in an efficient manner. The human intervention is reduced as the companies don't want to waste their valuable time of their employees in solving simple and redundant problems of their customers. So many companies use artificial intelligence in developing bots to solve their simple customer related queries. Companies often use bots to cut down on cost, instead of employing people to communicate with consumers, companies have developed new ways to be efficient. These chatbots are used to answer customers' questions.

In this project, we use the typical customer care chatbot experience but instead of relying on predefined responses, our dialog will provide a hook that can call out to other IBM Watson services for additional sources of information. In our case, it will be an owner’s manual that has been uploaded into Watson Discovery.

**Purpose:**

Chatbots are being made to ease the pain that the industries are facing today. The purpose of chat bots is to support and scale business teams in their relations with customers. It could live in any major chat applications like Facebook Messenger, Slack, Telegram, Text Messages, etc. They're useful in a business because they save time and effort by automating various aspects of customer support. However, the possibilities of chatbots aren't limited to answering customer questions, but also to collect customer related information.

**Literature review:**

**Existing problem:**

The typical customer care chatbot can answer simple questions, such as store locations and hours, directions, and maybe even making appointments. When a question falls outside of the scope of the pre-determined question set, the option is typically to tell the customer the question isn’t valid or offer to speak to a real person. This causes a discomfort to customers. Companies can cut down the expenses and time if they are able to handle these situations with automated bots.

**Proposed solution:**

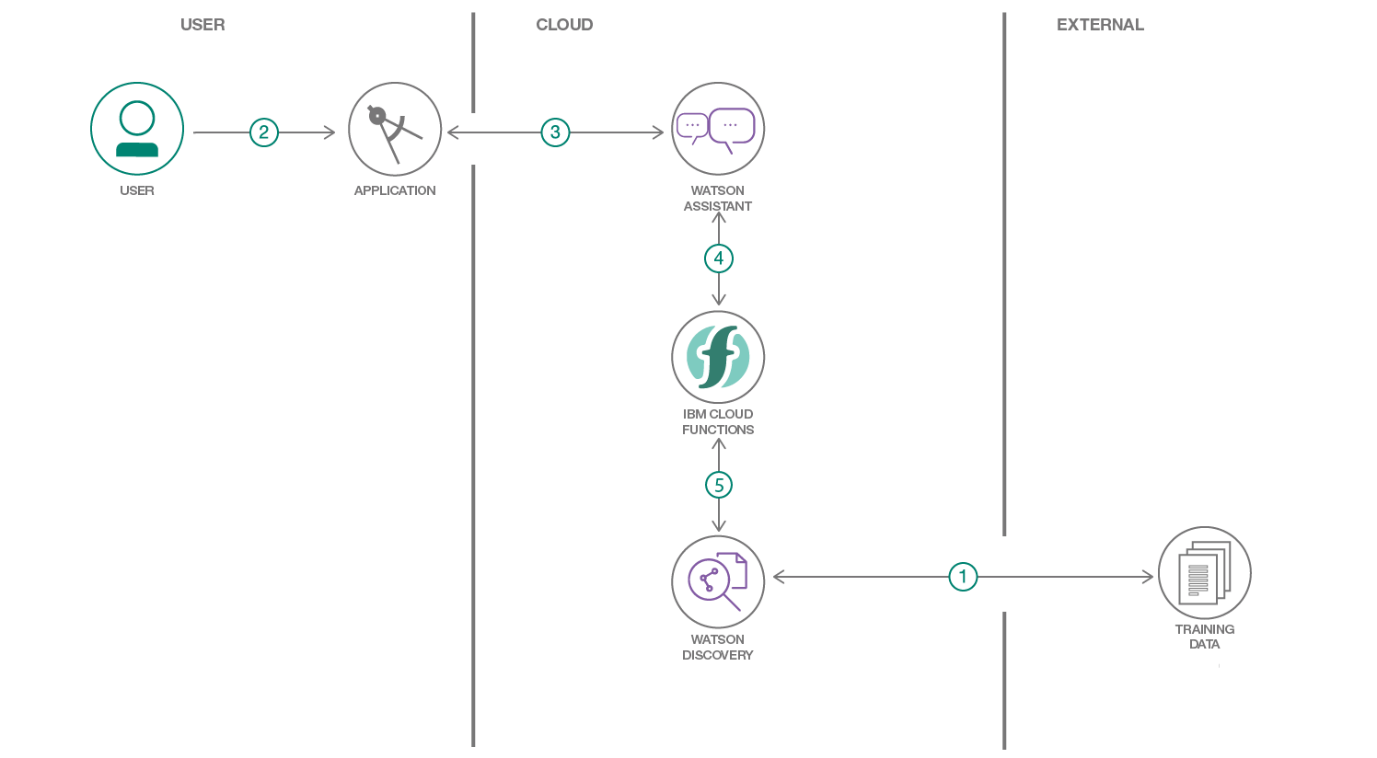
In this project, we will provide another option. If the customer question is about the operation of a device, we will use the webhook feature of Watson Assistant to pass the question onto our Watson Discovery Service, which has been pre-loaded with the device’s owner’s manual. So now, instead of “Would you like to speak to a customer representative?” we can return relevant sections of the owner’s manual to help solve our customers’ problems.

To take it a step further, the project shall use the Smart Document Understanding feature of Watson Discovery to train it on what text in the owner’s manual is important and what is not. This will improve the answers returned from the queries.

**Theoretical analysis:**

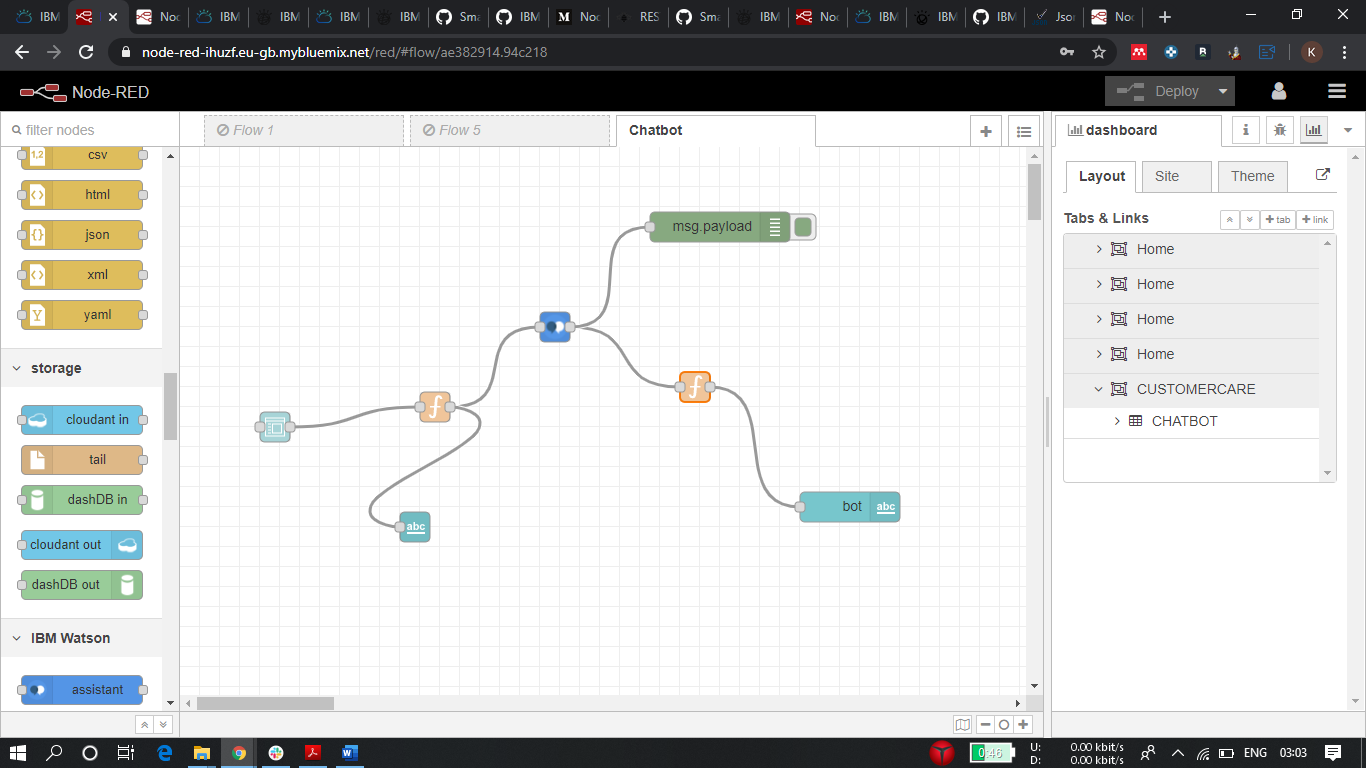
The AI bot will be trained with simple and basic questions. But if a complex question is asked like regarding the working of a product, the bot will offer to communicate with a real person. But here there will be a manual of the company is used by the bot to search and deliver relevant messages to the query. If an out of boundary question is asked, the bot invokes a cloud function to the Watson Discovery, which is preloaded with company's user manual. It then refers and delivers the relevant message to the customers via the bots.

**Block Diagram:**

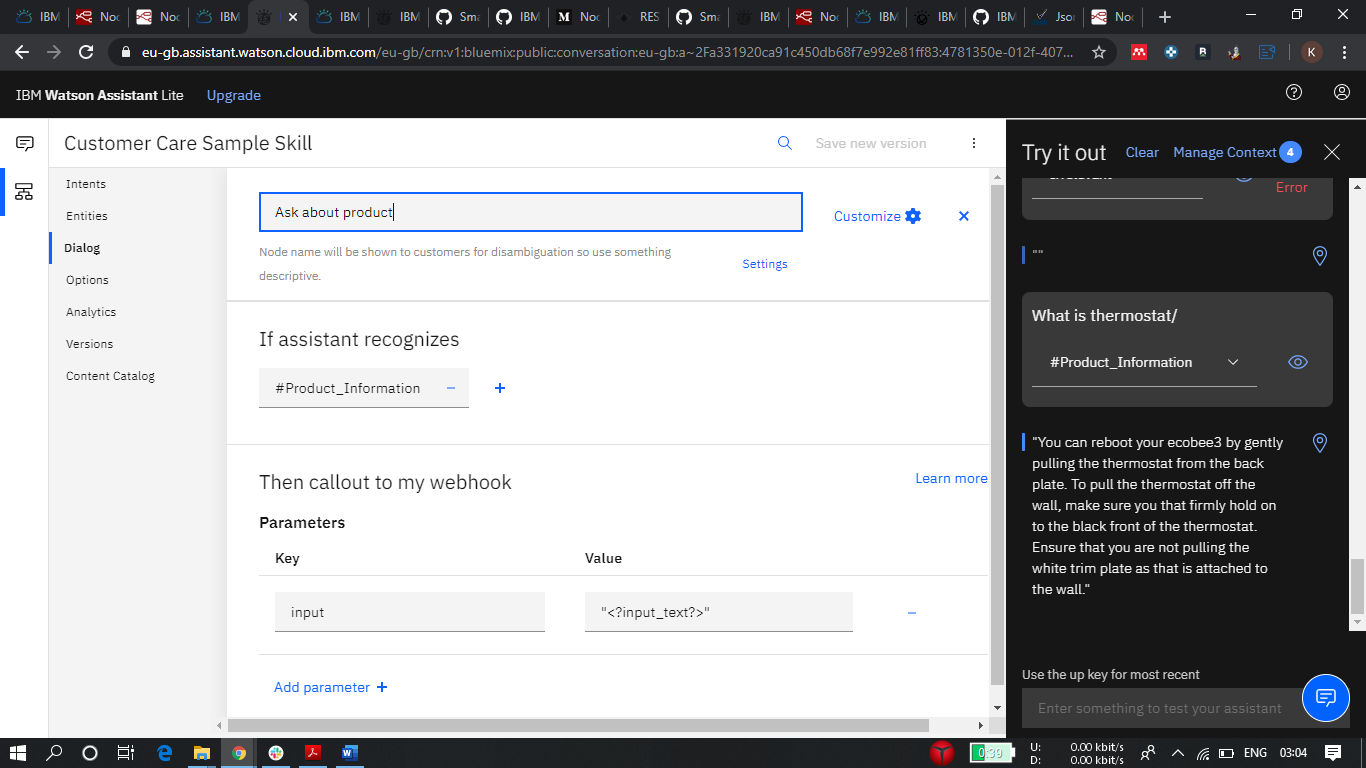
****

**Hardware/Software designing:**

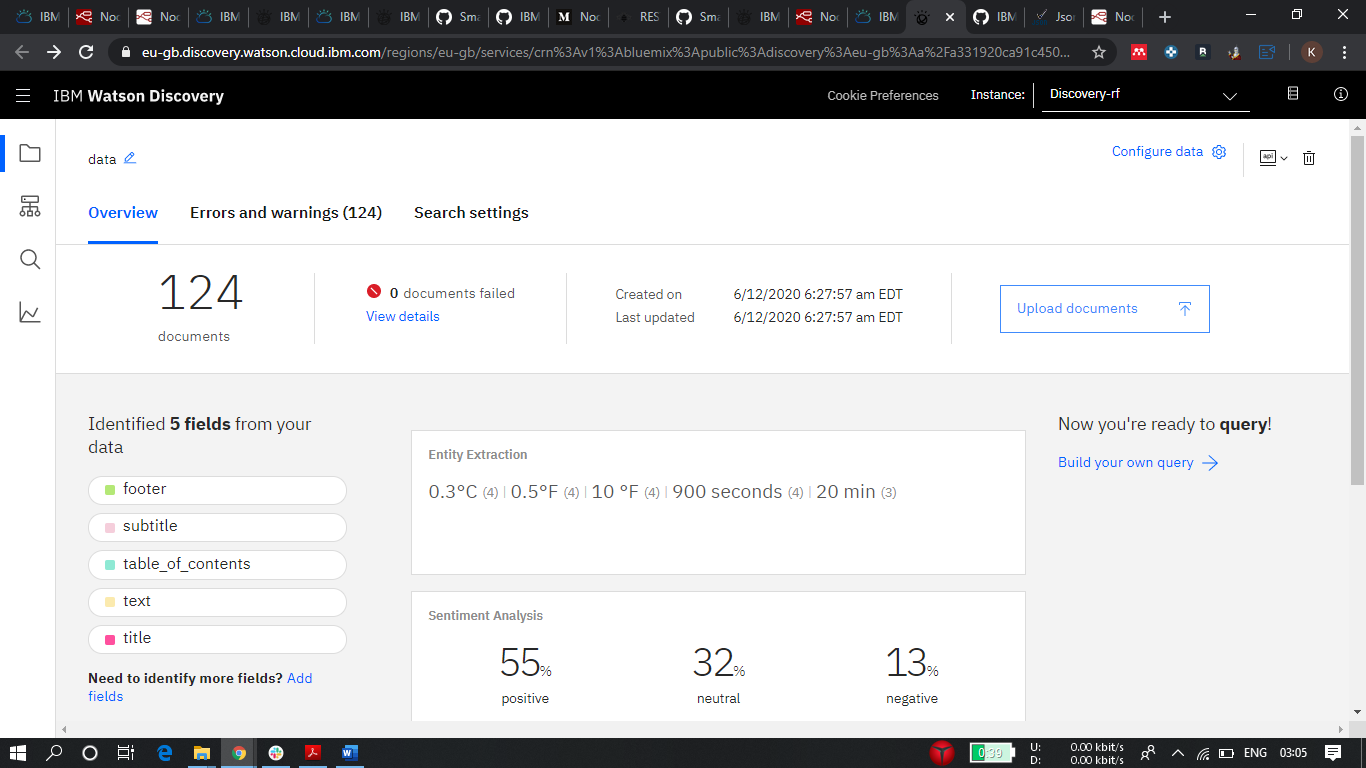
The chatbot is trained using IBM Watson Assistant. An IBM cloud function is created, which is used to invoke the Watson Discovery Service that uses smart documentation to answer queries from the user manual. IBM Watson Assistant invokes cloud function using webhook methodology. A node red application is used to integrate all these services and offer customers with ease of use.

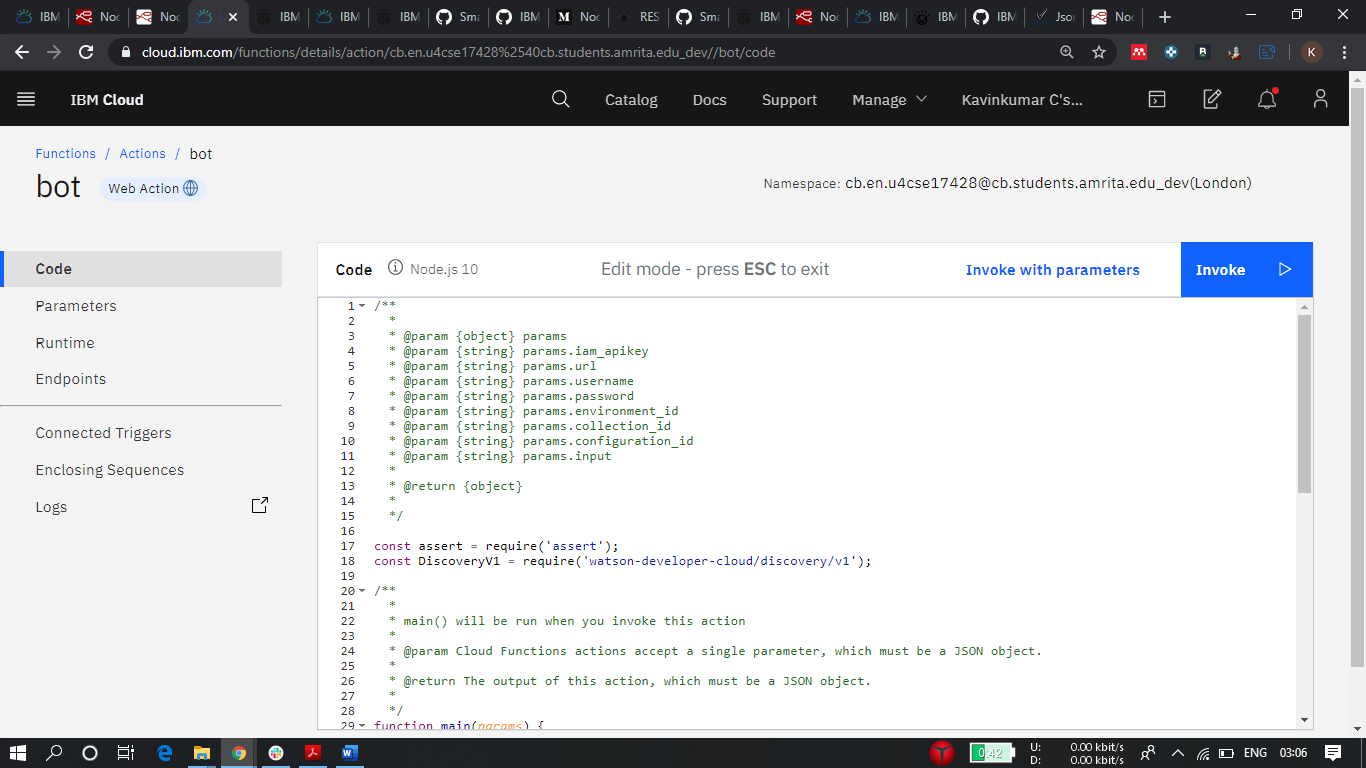
****

Node red application.

****

Watson assistant

****Watson Discovery

****

Cloud function

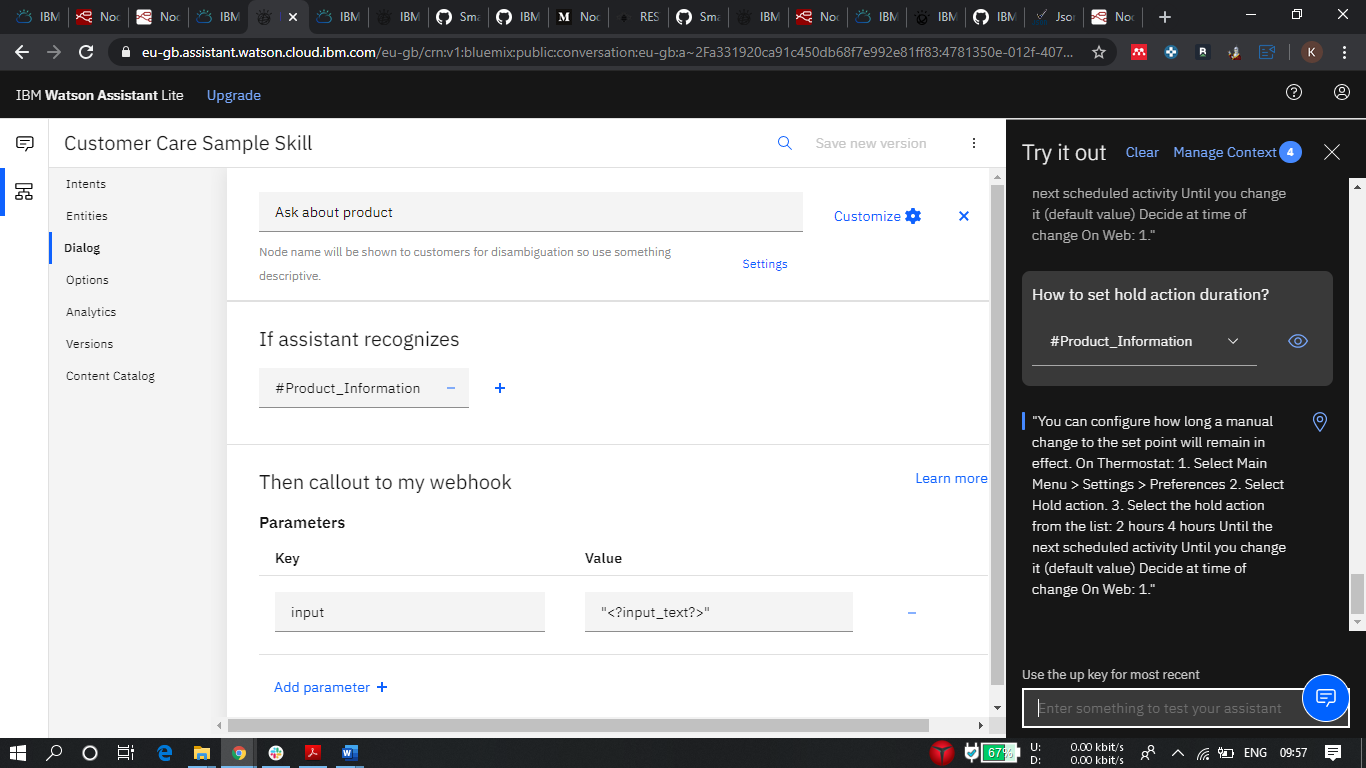
Smart Document Understanding leverages advances from IBM Research, as well as the company’s recently introduced Corpus Conversion System, an AI-based cloud service that can ingest 100,000 PDF pages per day (with accuracy above 97 percent) and then train and apply advanced machine learning models to extract content from the documents at scale.

SDU allows Watson Discovery customers to visually train AI to understand documents, to distinguish textual elements, to extract valuable information and to exclude “noise” like headers and footers. That’s impressive but in addition, no technical training is required for using SDU. Instead, a visual interface allows workers to point and click on elements such as titles, subtitles, headers and footers in training documents. The Watson system then displays how it understands the fields so staff can correct and resubmit documents if necessary.

**Experimental Investigations:**

First, I have created a node red application with respected nodes. Then I created a Watson assistant and trained a bot. Then I have created a discovery and populated with a user manual. Then a cloud function was created and integrated with Watson assistant and discovery. Now the node-red application is integrated with the bot, thus getting a flow which is shown in the flowchart below.

The below image shows the result for query, “How to set hold action duration?"

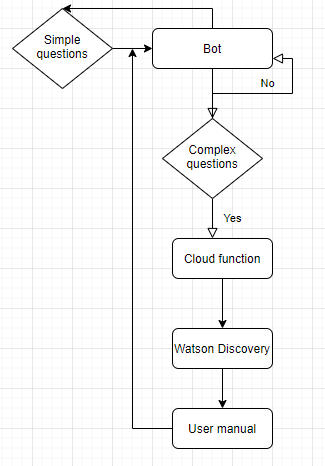
****

The chatbot retrieves only the relevant information regarding the query from the document and does not show any irrelevant information.

Businesses today have a lot of data—data in the cloud, data on premises, unstructured data, structured data. It’s not uncommon for enterprise AI training plans to contain terabytes of documents that must be ingested, annotated and the enrichments tested before useful results can be delivered. Search processes like that can consume a lot of resources before beginning to generate value. If only there was a way to make search intelligent.

Introducing, smart document understanding (SDU), a powerful feature now available in Watson Discovery, that can give you accurate answers faster, from 10 days to 2 minutes for one leading bank.

**Flowchart:**

****

1. The document is annotated using Watson Discovery Smart Document Understanding.

2. The user interacts with the back-end server via the app UI. The front-end app UI is a chatbot that engages the user in a conversation.

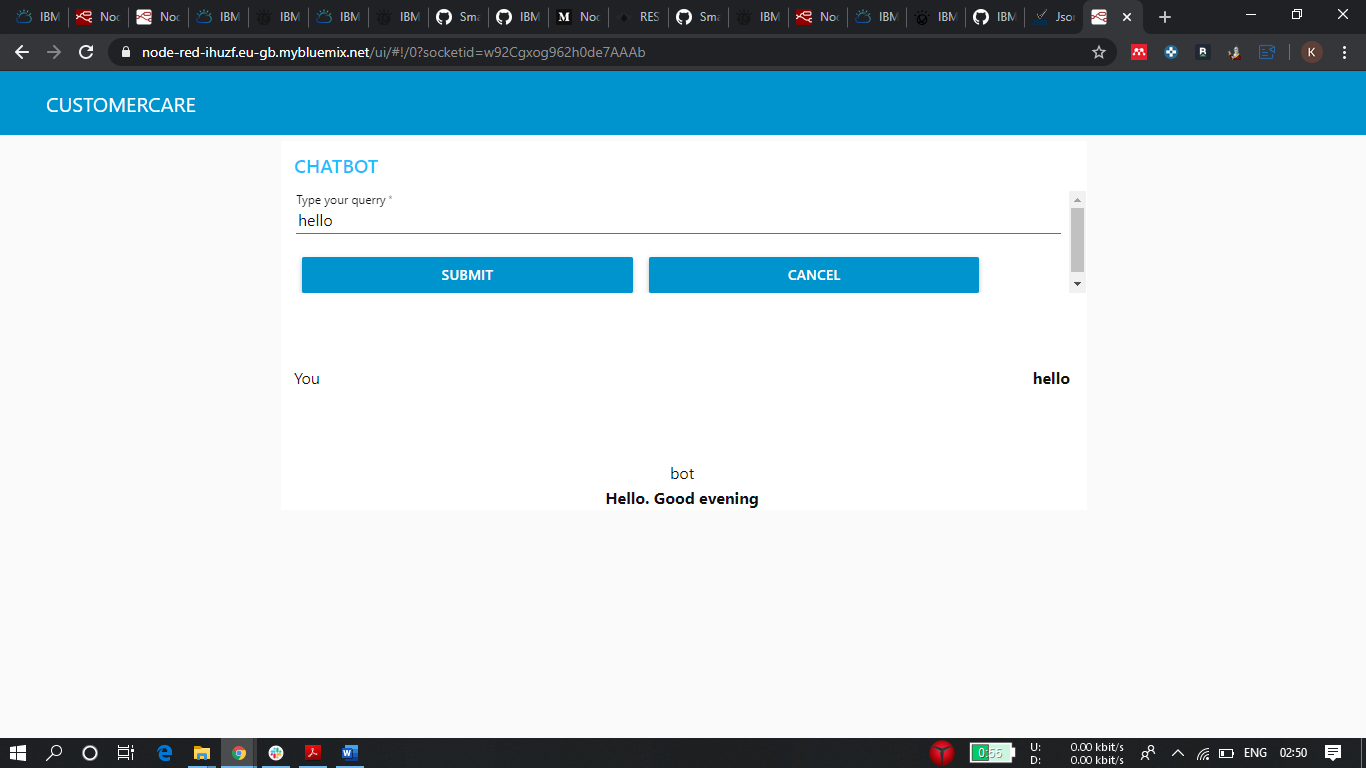
3. Dialog between the user and back-end server is coordinated using a Watson Assistant dialog skill.

4. If the user asks a product operation question, a search query is passed to a predefined IBM Cloud Functions action.

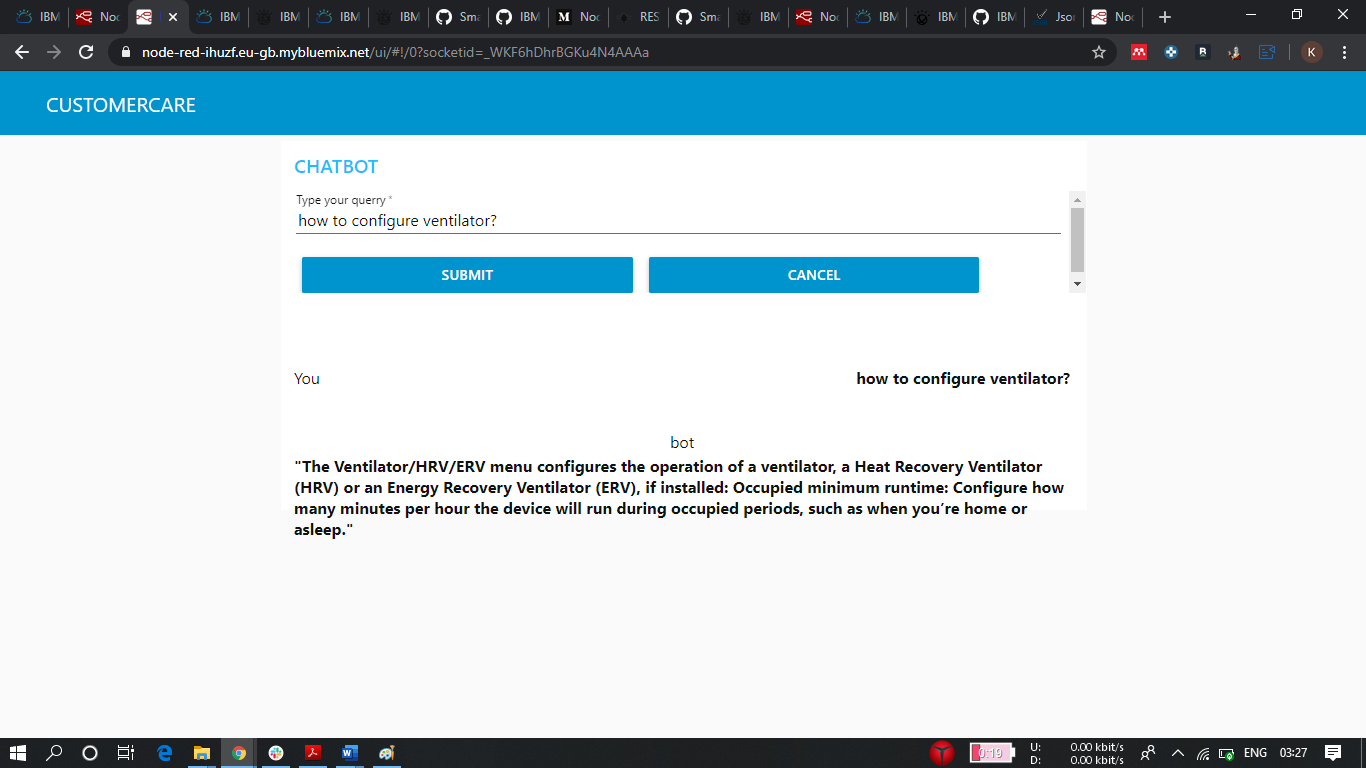
5. The IBM Cloud Functions action will query the Watson Discovery Service and return the results.

**Result:**

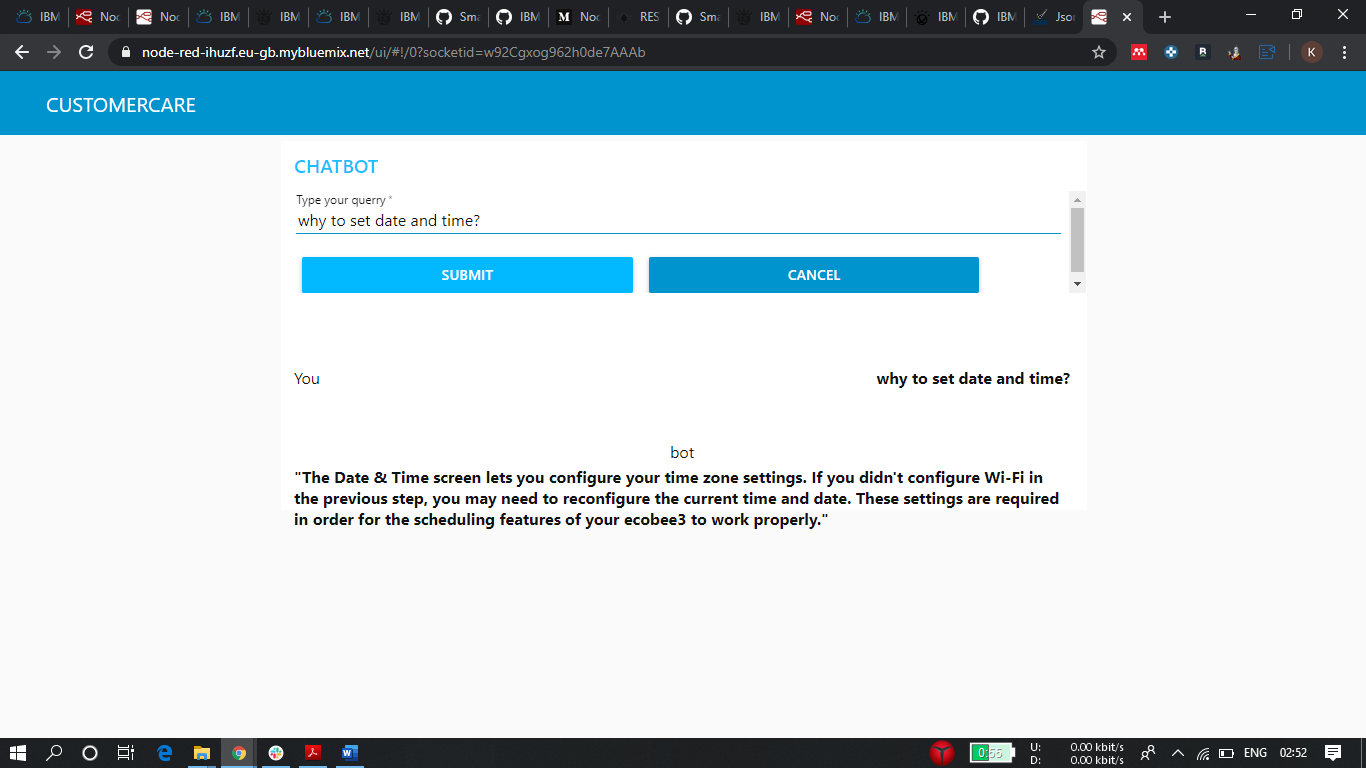
The bot was able to successfully retrieve the related information from the user manual when a question about the information of the product was asked. The following screenshots depict the situations where the bots were useful.

****Simple message to greet.

The output for the query "How to configure ventilator?" is given below

****

The output for the query "Why to configure date and time?" is given below

****

**Advantages and disadvantages:**

**Advantages:**

* Gather Customer Insights
* Increase Sales
* Available 24/7
* Save Money
* Improve Customer Satisfaction

**Disadvantages:**

* Lack Emotions
* Require Maintenance
* Bad training lead to irrelevant answers
* Customer Dissatisfaction in case of wrong details

**Applications:**

* Can be used by companies to deliver hardware related queries.
* Can be used by companies to deliver product information and configuration queries.
* Sales proposal creation for a Bank.
* Gives information on how to dismantle or reassemble a product of a specific company.
* Bank related queries.

**Conclusions:**

This project explained how to use the Smart Document Understanding feature of Watson Discovery to train the chatbot to call out to other IBM Watson services for additional sources of information.

**Future scope:**

The future scope of this project is to integrate voice assistant with the existing project. And also offer remote control system feature to customer by automated bots to solve their complex problems. This reduces the risk of customers to solve their problems and the bots will become more user-friendly.

**Bibliography:**

Kavinkumar C (2020), Amrita Vishwa Vidyapeetham

<https://www.linkedin.com/in/kavinkumar-c-1403b916a/>

**Appendix:**

**Source code (cloud function):**

/\*\*

\*

\* @param {object} params

\* @param {string} params.iam\_apikey

\* @param {string} params.url

\* @param {string} params.username

\* @param {string} params.password

\* @param {string} params.environment\_id

\* @param {string} params.collection\_id

\* @param {string} params.configuration\_id

\* @param {string} params.input

\*

\* @return {object}

\*

\*/

const assert = require('assert');

const DiscoveryV1 = require('watson-developer-cloud/discovery/v1');

/\*\*

\*

\* main() will be run when you invoke this action

\*

\* @param Cloud Functions actions accept a single parameter, which must be a JSON object.

\*

\* @return The output of this action, which must be a JSON object.

\*

\*/

function main(params) {

return new Promise(function (resolve, reject) {

let discovery;

if (params.iam\_apikey){

discovery = new DiscoveryV1({

'iam\_apikey': params.iam\_apikey,

'url': params.url,

'version': '2019-03-25'

});

}

else {

discovery = new DiscoveryV1({

'username': params.username,

'password': params.password,

'url': params.url,

'version': '2019-03-25'

});

}

discovery.query({

'environment\_id': params.environment\_id,

'collection\_id': params.collection\_id,

'natural\_language\_query': params.input,

'passages': true,

'count': 3,

'passages\_count': 3

}, function(err, data) {

if (err) {

return reject(err);

}

return resolve(data);

});

});

}

**Source code (node-red app):**

[{"id":"ae382914.94c218","type":"tab","label":"Chatbot","disabled":false,"info":""},{"id":"7458b012.3e783","type":"watson-conversation-v1","z":"ae382914.94c218","name":"Chatbot","workspaceid":"b6113320-3264-4835-b812-960f6ce67c0f","multiuser":false,"context":true,"empty-payload":false,"service-endpoint":"https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/4781350e-012f-407c-9c13-a86a4ea5e178","timeout":"","optout-learning":false,"x":375,"y":180,"wires":[["90ba8345.59105","af85dd0e.38f95"]],"l":false},{"id":"d7a16bfb.ca8568","type":"function","z":"ae382914.94c218","name":"F1","func":"msg.payload = msg.payload.text; \nreturn msg;","outputs":1,"noerr":0,"x":255,"y":260,"wires":[["7458b012.3e783","c4908762.c5c438"]],"l":false},{"id":"e56f148d.47ce38","type":"ui\_form","z":"ae382914.94c218","name":"","label":"","group":"22a93b98.c31e74","order":1,"width":0,"height":0,"options":[{"label":"Type your querry","value":"text","type":"text","required":true,"rows":null}],"formValue":{"text":""},"payload":"","submit":"submit","cancel":"cancel","topic":"","x":95,"y":280,"wires":[["d7a16bfb.ca8568"]],"l":false},{"id":"90ba8345.59105","type":"function","z":"ae382914.94c218","name":"F2","func":"if(msg.payload.output.error){\n msg.payload = \"please rephrase\";\n return msg;\n}\nmsg.payload = msg.payload.output.text[0];\nreturn msg;","outputs":1,"noerr":0,"x":515,"y":240,"wires":[["ede17173.a8dd7"]],"l":false},{"id":"c4908762.c5c438","type":"ui\_text","z":"ae382914.94c218","group":"22a93b98.c31e74","order":3,"width":0,"height":0,"name":"","label":"You","format":"{{msg.payload}}","layout":"row-spread","x":235,"y":380,"wires":[],"l":false},{"id":"ede17173.a8dd7","type":"ui\_text","z":"ae382914.94c218","group":"22a93b98.c31e74","order":5,"width":0,"height":0,"name":"","label":"bot","format":"{{msg.payload}}","layout":"col-center","x":670,"y":360,"wires":[]},{"id":"af85dd0e.38f95","type":"debug","z":"ae382914.94c218","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"false","x":540,"y":80,"wires":[]},{"id":"22a93b98.c31e74","type":"ui\_group","z":"","name":"CHATBOT","tab":"5e28dda6.14e9e4","order":1,"disp":true,"width":15,"collapse":false},{"id":"5e28dda6.14e9e4","type":"ui\_tab","z":"","name":"CUSTOMERCARE","icon":"","disabled":false,"hidden":false}]