PROJECT REPORT

***OVERVIEW:***

the typical customer care chat bot can answer simple questions , such as store locations and hours, directions,and maybe even making appoinments. when a question falls outside the scope of the pre determined question set, the option is typically to tell the customer the question is not valid or offer to speak to real person.

in this project, there will be another option. if the ccustomer question is about the operation of a device, the application shall pass the question on to watson discovery service,which has been preloaded with the device owner's manual. so now instead of"would you like to speakto a customer representative? we can return relevant sections of a owner's manual to solve customer's problems.

to take it a step further, the project shall use the smart document undestanding feature of watson discovery to train it on what text in owner's manual is important and what is not. this will improve the answers returned from the queries.

***PURPOSE:***

1. simplify the work of customer representative.
2. can reduce the workforce in the company.
3. with sdu feature the bot can provide more relevant information to the customer.
4. it is cost effective.
5. time saver
6. due to the chat bot can provide better service to their customers.

***LITERATURE REVIEW:***

***EXISTING PROBLEM:***

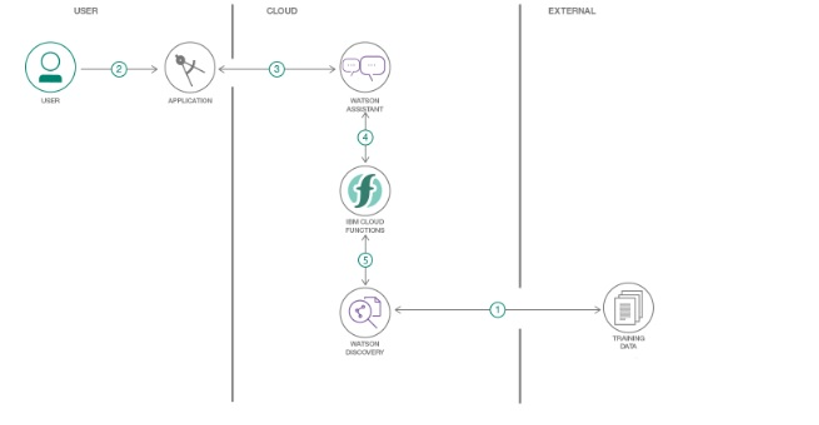
if the customer has any problem related to the device or if he does not know how to operate a device he has to call the company's customer care representative.there are chances that the representative is not able to answer the call in time or he is not trained enough to answer the question.even the company has to train the representative for the work and if the representative leaves the work the comapny has to train a new person for the same work.

***PROPOSED SOLUTION:***

the chat bot with smart document understanding can answer the customer's doubts and queries any time instead of directing the question to the customer care representative.the bot also needs to be trained only once ,this can save the company's resources and time for the company with the improvement in the company's services toward the customers.

THEORITICAL SURVEY:

***BLOCKDIAGRAM/SOFTWARE DESIGNING***:



***EXPERIMENTAL INVESTIGATION:***

1)IBM account was created.

2)watson assistant and watson discovery were the services created.

3)Node Red application was created.

4)Watson assistant was trained by creating necessary intents, entities and dialogs.

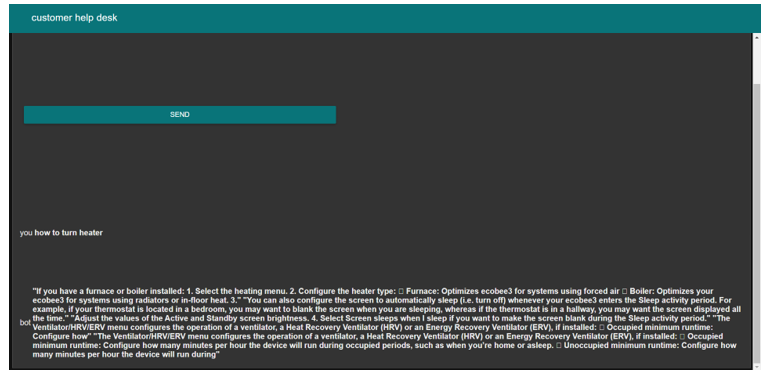
5)Watson discovery was used to configure the training dat(owner's manual) using the smart document understanding .the fields like identity fields and mange fields were used to divde the document in to subtitles,footers,text etc.

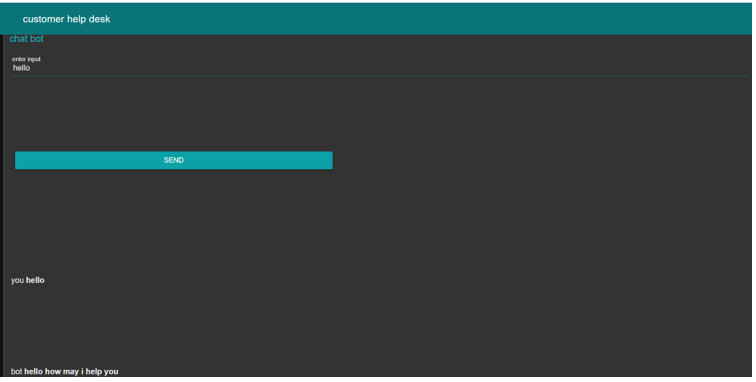
6)Cloud functions were created to enable a web action that allows Watson assistant to post queries to Watson discovery.

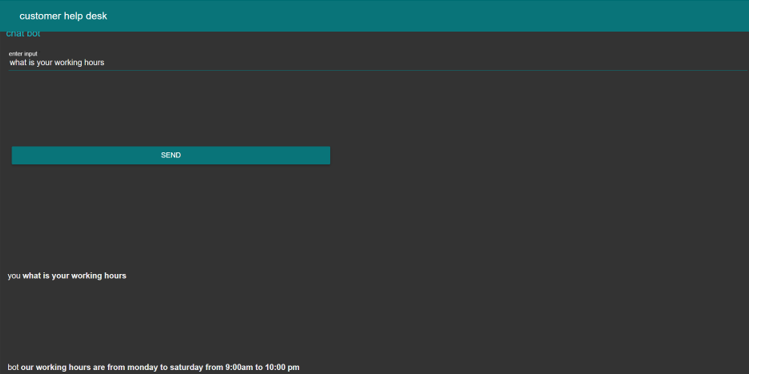
7)Node Red application was used to create a dashboard by integrating various services.

8)thus a user interface web application was created using node red.

***RESULTS:***









***ADVANTAGES:***

1)Cost effective.

2)It can reduce the work force for the company.

3)It can help the company to provide better services to the customers.

4)It can reduce the workload on the customer care representatives.

***DISADVANTAGES:***

1)The customers need to frame the questions properly or the results may be malicious.

2)The bot can provide the questions for only which the training data is fed thus limiting the range of answers it can provide.

***APPLICATIONS:***

The bot can be trained to provide a wide range of answers .More the training data fed to the bot, more the range of questions it can answer.It can be used as the customer care representative for the company. It can act as an office assistant.It can also be used in banks directing the customers how to fill the form etc. In the similar way the bot can be used in railways to direct the customers how to book the tickets and for enquiry.

***CONCLUSION:***

Thus we have created a bot which can answer various queries post by the user. It can be concluded that by creating IBMcloud functions web action that allows Watson assistant to post queries to Watson discovery and integrating the services using NodeRed a user interface web dashboard can be created.

***FUTURESCOPE:***

The more the training data fed to the bot the more the range of questions it can answer.The more diversely the data fed to the bot the more diversely for example by providing the weather api of the weather website to the bot , it can predict weather. As mentioned in the application section the bot can be used in the banks for helping the customers to fill the form and directing the customers the particular counters where the particular work is done.Even the bot can be used in Railways for enquiry and how to book tickets online.

***BIBLIOGRAPHY:***

<https://cloud.ibm.com/docs/overview?topic=overview-whatis-platform>

<https://www.youtube.com/embed/r7E1TJ1HtM0>

https://www.ibm.com/cloud/watson-assistant/docs-resources/

<https://developer.ibm.com/articles/introduction-watson-discovery/>

<https://cloud.ibm.com/docs/services/discovery?topic=discovery-getting-started>

<https://cloud.ibm.com/apidocs/functions>

<https://nodered.org/docs/getting-started/ibmcloud>

***APPENDIX:***

***source code:***

/\*\*

\* @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({

' I am\_apikey' : params.i am\_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);

});

});

}

***NODE RED FLOW:***