Project Report

on

Intelligent Customer Helpdesk with Smart Document Understanding

in

Artificial Intelligence

K SAI ABHINAV

(SPS_PRO_99)

Intelligent Customer Help Desk with Smart Document Understanding

1. Introduction

1.1 Overview

We will build a chatbot that uses IBM Services (Watson Discovery, Watson Assistant, Watson Cloud Functions and Node-Red) to deliver an effective Web based UI through which we can chat with the assistant.

We will integrate the Watson Discovery service with Watson Assistant using webhooks.

- Project Requirements: Node-RED, IBM Cloud, IBM Watson, Node JS, User Manual

- Functional Requirements: IBM Cloud

- Technical Requirements: AI, ML, Watson AI, Node JS

- Software Requirements: Watson Assistant, Watson Discovery, Watson Cloud Functions, Node-RED

- Project Deliverables: Intelligent Chatbot with Smart Document Understanding

- Project Team: Individual

- Project Duration: 28 Days

1.2   Purpose

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.

In this project, there will be another option. If the customer question is about the operation of a device, the application shall pass the question onto Watson Discovery Service, which has been pre-loaded with the device's owners manual. So now, instead of "Would you like to speak to a customer representative?" we can return relevant sections of the owners manual to help solve our customers' problems. So unless and until customer specifically asks for a customer representative the bot will try to solve all your queries.

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. Then using Watson actions as webhook, Watson Discovery can be integrated with Watson assistant. Finally using Node-Red, Watson assistant can be integrated with a web UI. This UI can then be used to connect with Watson assistant and chat with it.

1.2.1   Scope of Work

- Create a customer care dialog skill in Watson Assistant
- Use Smart Document Understanding to build an enhanced Watson Discovery collection
- Create an IBM Cloud Functions web action that allows Watson Assistant to post queries to Watson Discovery
- Build a web application with integration to all these services & deploy the same on IBM Cloud Platform

2.   Literature Survey

2.1   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.

2.2   Proposed Solution

In this project, there will be another option. If the customer question is about the operation of a device, the application shall pass the question onto Watson Discovery Service, which has been pre-loaded with the device's owners manual. So now, instead of "Would you like to speak to a customer representative?" we can return relevant sections of the owners manual to help solve our customers' problems. So unless and until customer specifically asks for a customer representative the bot will try to solve all your queries.

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 owners manual is important and what is not. This will improve the answers returned from the queries. Then using Watson actions as webhook, Watson Discovery can be integrated with Watson assistant. Finally using Node-Red, Watson assistant can be integrated with a web UI. This UI can then be used to connect with Watson assistant and chat with it.

3.   Theoretical Analysis
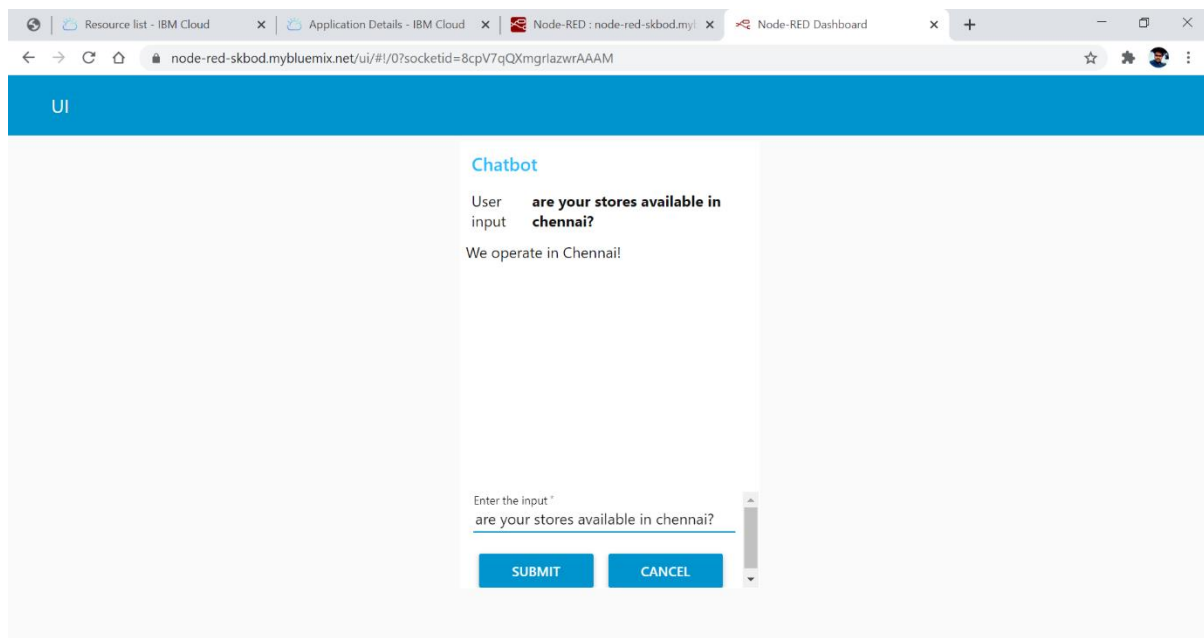
3.1   Block / Flow Diagram
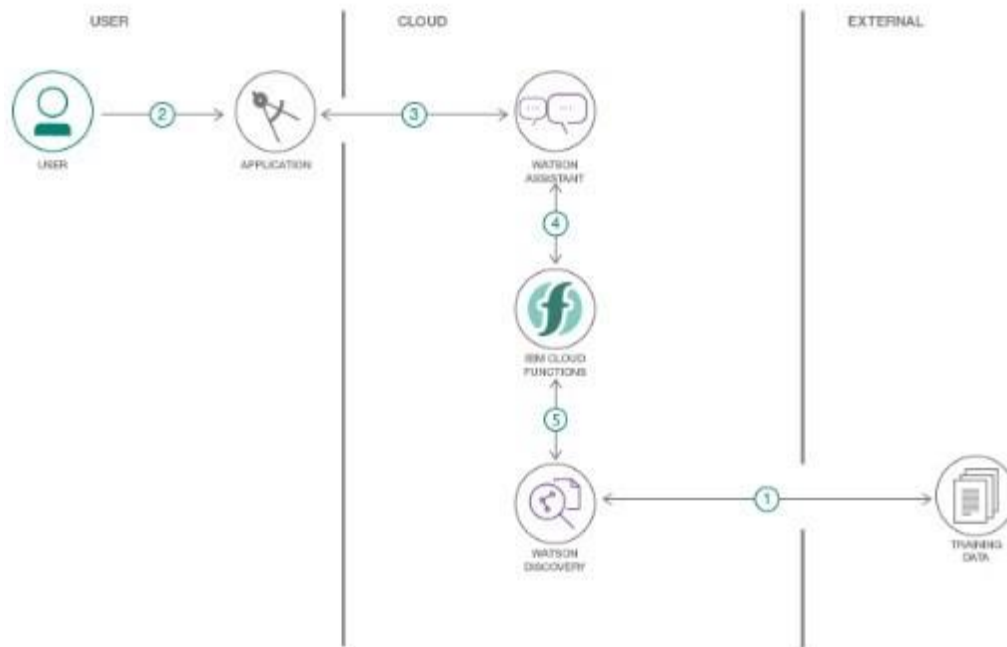
3.2   Hardware / Software Designing

1. Create necessary Watson Services.
2. Configure Watson Discovery.
3. Create Watson Cloud Functions Action.
4. Configure Watson Assistant.
5. Integrate Watson Discovery with Watson Assistant using webhook.
6. Build Node-RED flow to integrate Watson Assistant and Web Dashboard.

4.   Experimental Investigation

Chatbot

User input **what are the operating timings in chennai?**

We are open from 10 A.M and close at 8 P.M everyday.

Enter the input *
what are the operating timings in chenr

SUBMIT   CANCEL

Chatbot

User input **how to turn on the speaker?**

"1. Press and hold (2 secs) the Bluetooth button on the Jabra Speak 510 until the Bluetooth light flashes blue. Release the button when you hear the connection instructions. 2. Plug the Jabra Link 370 into any available USB port on your computer. 3. The connection is established when the light turns solid blue. NOTE: Ensure that all Bluetooth devices previously connected to the Jabra Link 370 are switched off."

Enter the input *
how to turn on the speaker?

SUBMIT   CANCEL

UI

Chatbot

User input **how to change between devices?**

"1. Press and hold the Bluetooth button and tap the Volume up or Volume down button to cycle between the connected devices. The connected devices will be announced. 2. Release the Bluetooth button when the correct device is announced. The Jabra Speak 510 will establish a connection with the device(s). When a connection has been established the device(s) will be announced."

Enter the input *
how to change between devices?

SUBMIT   **CANCEL**

UI

Chatbot

User input **how to charge the speaker?**

"Plug the Jabra Speak 510 USB cable into any available USB port on your computer to charge the battery. The battery light is red when charging, and green for 1 minute when fully charged."

Enter the input *
how to charge the speaker?

SUBMIT   **CANCEL**

Chatbot

User input **how to connect to jabra link**

"1. Press and hold (2 secs) the Bluetooth button on the Jabra Speak 510 until the Bluetooth light flashes blue. Release the button when you hear the connection instructions. 2. Plug the Jabra Link 370 into any available USB port on your computer. 3. The connection is established when the light turns solid blue. NOTE: Ensure that all Bluetooth devices previously connected to the Jabra Link 370 are switched off."

Enter the input *
how to connect to jabra link

SUBMIT   CANCEL

Chatbot

User input   **thanks for your help**

Always there to help you!

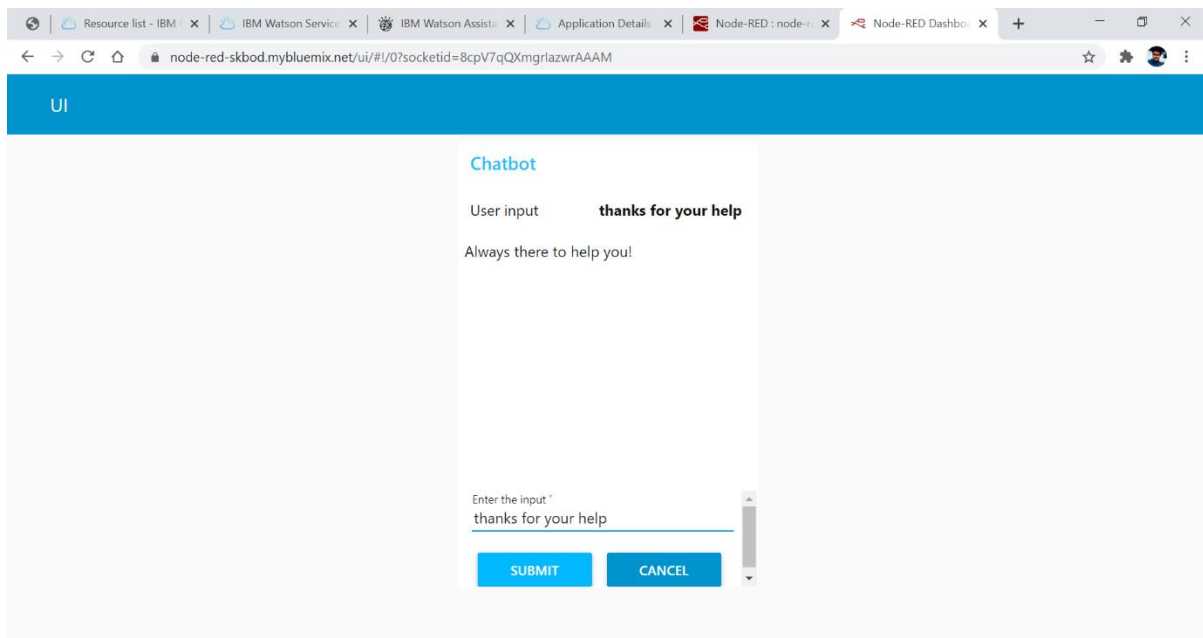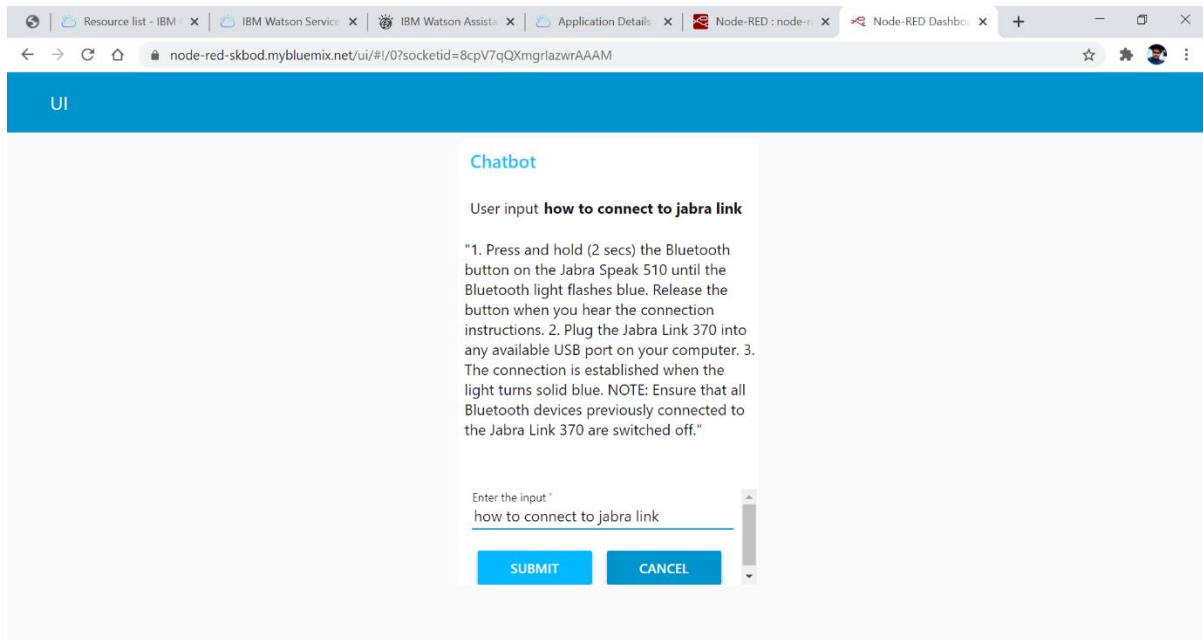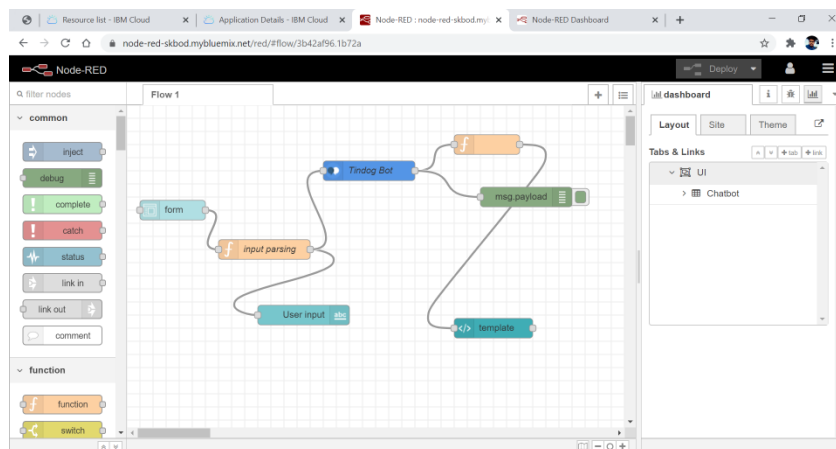Enter the input *
thanks for your help

SUBMIT   CANCEL

5.   Flowchart

Insert the following nodes into the flow in Node-RED.

- Inject
- ui_form
- Function (Input function and output function)
- ui_text
- Watson-conversation-v1 (Watson assistant node)
- Debug (Assistant debug and output function debug)
- ui_template

6.   Results

Web based UI was developed by integrating all the services using Node-RED.

URL for UI Dashboard : https://node-red-skbod.mybluemix.net/ui/

7.   Advantages & Disadvantages

Advantages

1. Reduces Man Power
2. Cost Efficient
3. Less and less calls will be diverted to Customer Representatives.

Disadvantages

1. Sometimes it can mislead customers as it tries to search irrelevant information in the manual.
2. It may also give same answers to different queries.

8. Applications

1. This chatbot can be deployed to various websites as it can solve a lot of basic questions.

2. It can be used to deploy as Customer Helpdesk for small scale products as their manual usually has the solution for the user's problems.

9. Conclusion

An Intelligent Customer Helpdesk Chatbot was created using various Watson services like Watson Discovery, Watson Assistant, Watson Cloud Functions and Node-RED.

10. Future Scope

In the future, various other Watson services like Text-To-Speech and Speech-To-Text can be integrated in the chatbot. This can make the chatbot Hands-free.

11. Bibliography

1. Node-RED Starter Application: https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/

2. Build your open AI assistant: https://www.youtube.com/watch?v=hitUOFNne14

3. How to use Watson Assistant with Webhooks: https://www.youtube.com/embed/5z3i5IsBVnk

4. Watson Discovery: https://cloud.ibm.com/docs/discovery?topic=discovery-getting-started

5. Smart document understanding in Watson discovery: https://www.youtube.com/watch?v=-yniuX-Poyw&feature=youtu.be

Appendix

Source Code

Watson Cloud Function Action 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({
        '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);
            });
         });
      }
```

Node-RED Flow code

```
   [

1. {
o   "id":"3b42af96.1b72a",
o   "type":"tab",
o   "label":"Flow 1",
o   "disabled":false,
o   "info":""

   },

2. {
o   "id":"452c15d8.337dfc",
o   "type":"ui_form",
o   "z":"3b42af96.1b72a",
o   "name":"",
o   "label":"",
o   "group":"9c0aa41c.e60498",
o   "order":3,
o   "width":"0",
o   "height":"0",
o   "options":[
1. {
▪   "label":"Enter the input",
▪   "value":"input",
▪   "type":"text",
▪   "required":true,
```

- "rows":*null*

    }

o],
o "formValue":{
- "input":""

    },

o "payload":"",
o "submit":"submit",
o "cancel":"cancel",
o "topic":"",
o "x":70,
o "y":160,
o "wires":[
0. [
1. "e686e2ab.8d501"

    ]

o]
3. },
4. {
o "id":"e686e2ab.8d501",
o "type":"function",
o "z":"3b42af96.1b72a",
o "name":"input parsing",
o "func":"msg.payload = msg.payload.input;\nreturn msg;",
o "outputs":1,
o "noerr":0,
o "x":210,
o "y":220,
o "wires":[
0. [
0. "4377bc73.b41d94",
1. "112a93e9.672dbc"

    ]

o]
5. },
6. {
o "id":"4377bc73.b41d94",
o "type":"ui_text",
o "z":"3b42af96.1b72a",
o "group":"9c0aa41c.e60498",
o "order":1,
o "width":"6",
o "height":"1",
o "name":"",

- o "label":"User input",
- o "format":"{{msg.payload}}",
- o "layout":"row-spread",
- o "x":270,
- o "y":320,
- o "wires":[

  ]

7. },
8. {
- o "id":"54021493.af868c",
- o "type":"debug",
- o "z":"3b42af96.1b72a",
- o "name":"",
- o "active":true,
- o "tosidebar":true,
- o "console":false,
- o "tostatus":false,
- o "complete":"payload",
- o "targetType":"msg",
- o "x":610,
- o "y":140,
- o "wires":[

  ]

9. },
10. {
- o "id":"b75bba58.e182f8",
- o "type":"function",
- o "z":"3b42af96.1b72a",
- o "name":"",
- o "func":"msg.payload = msg.payload.output.text[0]\nreturn msg;",
- o "outputs":1,
- o "noerr":0,
- o "x":550,
- o "y":60,
- o "wires":[
0. [
0. "c37a63b5.65bbe"

  ]

- o]
11.   },
12. {
- o "id":"c37a63b5.65bbe",
- o "type":"ui_template",
- o "z":"3b42af96.1b72a",
- o "group":"9c0aa41c.e60498",

- "name":"",
- "order":2,
- "width":"6",
- "height":"5",
- "format":"<!DOCTYPE html>\n<html>\n<head>\n<style>\ndiv.scrl {\n background-color: white;\n width: 310px;\n height: 400px;\n overflow: auto;\n \n margin-bottom:20px;\n}\n</style>\n</head>\n<body>\n<div class=\"scrl\">{{msg.payload}}</div>\n</body>\n</html>\n",
- "storeOutMessages":true,
- "fwdInMessages":true,
- "resendOnRefresh":true,
- "templateScope":"local",
- "x":560,
- "y":340,
- "wires":[

0. [

    ]

-]

13.   },
14. {
- "id":"112a93e9.672dbc",
- "type":"watson-conversation-v1",
- "z":"3b42af96.1b72a",
- "name":"Tindog Bot",
- "workspaceid":"8875b156-4081-4ac5-945d-a0e97a8c34be",
- "multiuser":false,
- "context":true,
- "empty-payload":false,
- "service-endpoint":"https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/b8e26908-b856-4ca0-aa40-c17279ba2d67",
- "timeout":"",
- "optout-learning":false,
- "x":370,
- "y":100,
- "wires":[

0. [
0. "b75bba58.e182f8",
1. "54021493.af868c"

    ]

-]

15.   },
16. {
- "id":"9c0aa41c.e60498",
- "type":"ui_group",
- "z":"",
- "name":"Chatbot",

- o   `"tab":"43761bff.7481d4",`
- o   `"order":1,`
- o   `"disp":true,`
- o   `"width":"6",`
- o   `"collapse":false`

    `},`

17. `{`
- o   `"id":"43761bff.7481d4",`
- o   `"type":"ui_tab",`
- o   `"z":"",`
- o   `"name":"UI",`
- o   `"icon":"dashboard",`
- o   `"disabled":false,`
- o   `"hidden":false`

    `}`

    `]`