

**PROJECT REPORT**

**ON**

**INTELLIGENT CUSTOMER  
HELPDESK WITH SMART  
DOCUMENT UNDERSTANDING**

**IN**

**MACHINE LEARNING**

**BY**

**DEVASHISH TIWARI**  
**(devashishakatiwari@gmail.com)**

**MAY-JUNE 2020**

# **CONTENTS**

<b>1</b>	<b>INTRODUCTION</b>
	1.1 Overview
	1.2 Purpose
<b>2</b>	<b>LITERATURE SURVEY</b>
	2.1 Existing problem
	2.2 Proposed solution
<b>3</b>	<b>THEORITICAL ANALYSIS</b>
	3.1 Block diagram
	3.2 Hardware / Software designing
<b>4</b>	<b>EXPERIMENTAL INVESTIGATIONS</b>
<b>5</b>	<b>FLOWCHART</b>
<b>6</b>	<b>RESULT</b>
<b>7</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>
<b>8</b>	<b>APPLICATIONS</b>
<b>9</b>	<b>CONCLUSION</b>
<b>10</b>	<b>FUTURE SCOPE</b>
<b>11</b>	<b>BIBILOGRAPHY</b>
	<b>APPENDIX</b>
	A. Source code

# 1. INTRODUCTION

## 1.1 OVERVIEW

We will design Customer care chat bot that can not only answer the basic questions such as store locations and hours, directions, making appointments but also could give the answers based on Watson Discovery service, which has been pre-loaded with the device manual.

- Project Requirements: Python, IBM Cloud, IBM Watson
- Functional Requirements: IBM Cloud
- Functional Requirements: Basic working knowledge of chatbots, understanding of programming, IBM Cloud
- Technical Requirements: Friendly with Python Programming Language, AI, ML, IBM Cloud, IBM Watson, Github and Git, Node JS.
- Software Requirement: Watson Assistant, Watson Discovery, Watson Cloud Functions, NODE-RED
- Project Deliverables: Intelligent Customer Helpdesk with Smart Document Understanding.
- Project Team: Devashish Tiwari
- Project Duration: 19 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.

## **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 predetermined question set, the option is typically to tell the customer the question isn't valid or offer to speak to a real person.

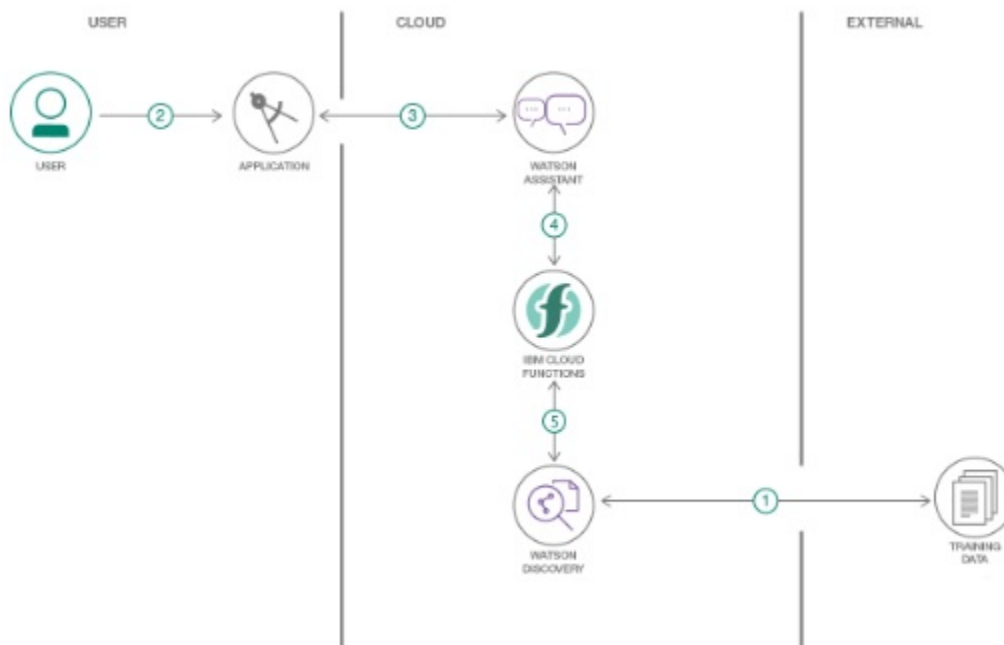
### **2.2 PROPOSED SOLUTION**

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 IBM Cloud services
2. Configure Watson Discovery
3. Create IBM Cloud Functions action
4. Configure Watson Assistant
5. Create flow and configure node
6. Deploy and run Node Red app

## 4. EXPERIMENTAL INVESTIGATIONS

Intelligent Customer Help Desk with Smart Document Understanding

Chatbot

You : **hello**

Enter your Input  
hello

SUBMIT

CANCEL

Bot : **Hello. Good afternoon**

Intelligent Customer Help Desk with Smart Document Understanding

Chatbot

You : **how to start heater**

Enter your Input  
how to start heater

SUBMIT

CANCEL

Bot : **"Smart recovery lets your ecobee3 learn how your heating and cooling system works, taking into account weather and historical operating performance so that your home reaches the scheduled set point at the time in which the change occurs (i.e. not afterwards). For example, if you wake up at 6:00 AM, you do not need to schedule your Home period to start at 5:30 AM. Smart Recovery will start the HVAC"**

## Intelligent Customer Help Desk with Smart Document Understanding

### Chatbot

You : **when to start heater**

Enter your Input

when to start heater

SUBMIT

CANCEL

Bot : "Specify what the heat pump runs when the O/B Reversing Valve is engaged: On Cool runs cooling when O/B engages (most cases), or On Heat runs heating when O/B engages. 4. Touch Next. You will be returned to the Equipment configuration menu."

## Intelligent Customer Help Desk with Smart Document Understanding

### Chatbot

You : **goodbye**

Enter your Input

goodbye

SUBMIT

CANCEL

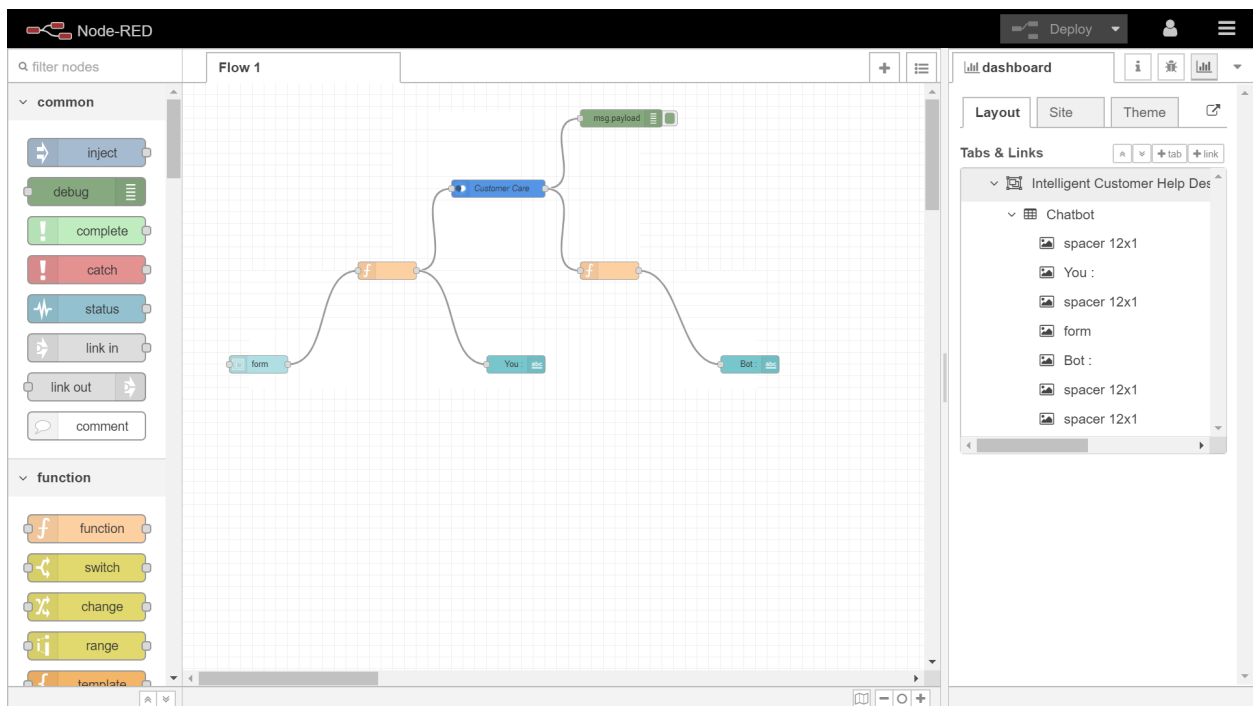
Bot : **So long**



## 5. FLOWCHART

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

- Debug
- ui\_Form
- ui\_Text
- Function
- Assistant



## 6. RESULT

Web based UI was developed by integrating all the services using Node-RED. URL for UI Dashboard :

<https://node-red-xihhb.eu-gb.mybluemix.net/ui/#!/0?socketid=CRrnnROyLTieHlj3AAAb>

## 7. ADVANTAGES & DISADVANTAGES

### Advantages:

- Companies can deploy chatbots to rectify simple and general human queries.
- Reduces man power .
- Cost efficient.
- No need to divert calls to customer agents and customer agents can look at other works.

### Disadvantages:

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

## 8. APPLICATIONS

- This chatbot can be deployed to various websites as it can solve a lot of basic questions.
- 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.
- The primary function of the chatbot is to be a virtual companion – To speak with senior people on general topics like the weather, nature, hobbies, movies, music, news, etc.

## 9. CONCLUSION

By doing the above procedure and all we successfully created an Intelligent help desk smart chatbot using Watson assistant, Watson discovery, Node-RED and cloud-functions.

## 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. BIBILOGRAPHY

- <https://www.ibm.com/cloud/get-started>
- <https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/>
- <https://github.com/watson-developer-cloud/node-red-labs>
- <https://www.youtube.com/embed/s7wmiS2mSXY>
- <https://www.youtube.com/watch?v=hitUOFNne14>
- <https://developer.ibm.com/components/watson-assistant/series/learning-path-watson-assistant>
- <https://developer.ibm.com/articles/introduction-watson-discovery/>
- <https://cloud.ibm.com/docs/openwhisk?topic=cloud-functions-getting-started>

# APPENDIX

## A. SOURCE CODE

### Node-RED Flow Code

```
[
  {
    "id": "e3c50f6f.0f43",
    "type": "tab",
    "label": "Flow
1",
    "disabled": false,
    "info": "",
    {
      "id": "7f5833de.574c3c",
      "type": "ui_form",
      "z": "e3c50f6f.0f43",
      "name": "",
      "label": "",
      "group": "216fd3f8.db046c",
      "order": 1,
      "width": 0,
      "height": 0,
      "options": {
        "label": "Enter your
input",
        "value": "text",
        "type": "text",
        "required": true,
        "rows": null
      }
    },
    {
      "formValue": {
        "text": ""
      },
      "payload": "",
      "submit": "submit",
      "cancel": "cancel",
      "topic": "",
      "x": 130,
      "y": 480,
      "wires": [
        [
          "25db5fc1.a30ee"
        ]
      ]
    },
    {
      "id": "25db5fc1.a30ee",
      "type": "function",
      "z": "e3c50f6f.0f43",
      "name": "",
      "func": "msg.payload
=msg.payload.text;\nreturn
msg;",
      "outputs": 1,
      "noerr": 0,
      "x": 350,
      "y": 320,
      "wires": [
        [
          "466adaa8.4516f4",
          "60a88535.42087c"
        ]
      ]
    },
    {
      "id": "f1805ec7.c33be",
      "type": "function",
      "z": "e3c50f6f.0f43",
      "name": "",
      "func": "msg.payloa
d=msg.payload.output.text[0];\nreturn
msg;",
      "outputs": 1,
      "noerr": 0,
      "x": 730,
      "y": 320,
      "wires": [
        [
          "cfb8290d.3132d8"
        ]
      ]
    },
    {
      "id": "466adaa8.4516f4",
      "type": "watson-conversation-v1",
      "z": "e3c50f6f.0f43",
      "name": "Customer
Care",
      "workspaceid": "22416c53-2ce4-4051-8947-fc709d531b24",
      "multiuser": false,
      "contex
t": true,
      "empty-payload": false,
      "service-endpoint": "https://api.eu-gb.assistant.watson.clou
d.ibm.com/instances/49c7ca87-7575-49ff-861e-996ef479e615",
      "timeout": "",
      "optout-lear
ning": false,
      "x": 540,
      "y": 180,
      "wires": [
        [
          "693087a6.b81ed8",
          "f1805ec7.c33be"
        ]
      ]
    },
    {
      "id": "60a88535.42087c",
      "type": "ui_text",
      "z": "e3c50f6f.0f43",
      "group": "216fd3f8.db046c",
      "order": 2,
      "width": 0,
      "height": 0,
      "name": "",
      "label": "You",
      "format": "{{msg.payload}}",
      "layout": "row-left",
      "x": 570,
      "y": 480,
      "wires": [
        [
          "693087a6.b81ed8"
        ]
      ]
    },
    {
      "id": "693087a6.b81ed8",
      "type": "debug",
      "z": "e3c50f6f.0f43",
      "name": "",
      "active": t
true,
      "tosidebar": true,
      "console": false,
      "tostatus": false,
      "complete": "false",
      "x": 750,
      "y": 60,
      "wires": [
        [
          "cfb8290d.3132d8"
        ]
      ]
    },
    {
      "id": "cfb8290d.3132d8",
      "type": "ui_text",
      "z": "e3c50f6f.0f43",
      "group": "216fd3f8.db046c",
      "order": 3,
      "width": 12,
      "height": 3,
      "name": "",
      "label": "Bot",
      "format": "{{msg.payload}}",
      "layout": "col-
center",
      "x": 970,
      "y": 480,
      "wires": [
        [
          "216fd3f8.db046c"
        ]
      ]
    },
    {
      "id": "216fd3f8.db046c",
      "type": "ui_group",
      "z": "",
      "name": "Ch
atbot",
      "tab": "7e347188.e32b9",
      "order": 1,
      "disp": true,
      "width": 12,
      "collapse": false
    },
    {
      "id": "7e347188.e32b9",
      "type": "ui_tab",
      "z": "",
      "name": "Customer Care
Helpdesk",
      "icon": "dashboard",
      "disabled": false,
      "hidden": false
    }
  ]
}
```

]

## 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);  
  
});
```

```
});
```

```
}
```