

**INTELLIGENT CUSTOMER HELP DESK WITH SMART DOCUMENT UNDERSTAND**

**NAME : SHAIK KALAM**

**CATEGORY : MACHINE LEARNING**

**MAIL ID : 160418737030@mjcollege.ac.**

**COLLEGE : MUFFAKHAM JHAH COLLEGE OF ENGINEERING AND TECHNOLOGY**

**INTERNSHIP AT smartinternz@2020**

## **1.INTRODUCTION**

- a.overview
- b.Purpose

## **2. LITERATURE SURVEY**

- a.existing problem
- b.Proposed solution

## **3.THEORETICAL ANALYSIS**

- a.blockdiagram
- b.software/hardware designing

## **4.EXPERIMENTAL INVESTIGATIONS**

### **5.FLOWCHART**

### **6.RESULT**

### **7.ADVANTAGES & DISADVANTAGES**

## **8.APPLICATIONS**

## **9.CONCLUSION**

## **10.FUTURE SCOPE**

## **11.BIBLIOGRAPHY**

## **APPENDIX**

- a.Sourcecode
- b.Reference

## **1.INTRODUCTION**

**a.Overview.**Here we are going to prepare a chatbot by the use of IBM CLOUD services and the WATSON services like ASSISTANT and DISCOVERY.

service we are using in making chat bot are Discovery,cloud function,assistant,Node-red.By the end of this project we will learn how to make the chat bot and best practices of integration of watson services and how they can build interactive information retrival systems with Discovery and Assitant with the combainations of functioons and integration with Node-red.

- i. **PROJECT REQUIREMENTS** : PYTHON,IBM CLOUD, IBM WATSON.
- ii. **FUNCTIONAK REQUIREMENTS** : IBM Cloud
- iii. **TECHNICAL REQUIEMENTS** : AI,ML,WATSON,PYTHON
- iv. **SOFTWARE REQUIREMENTS** : WATSON ASSISTANT,WATSON-DISCOVERY,NODE-RED
- v. **PROJECT DELIVERABLES** : Smartinternz internship
- vi. **PROJECT TEAM** : SHAIK KALAM
- vii. **PROJECT DURATION** : 19 DAYS

**b.purpose:**

the typical customer care chatbot can answer simple questions, such as store locations and hours, directions, and may be even making appointments or bookings mainly. When the question falls out 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 operation of device, then the application shall pass to the Watson Discovery service which has been preloaded with device's owner 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 customer.

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.

**i. Scope of work**

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

## 2.LITERATURE SURVEY

### **a.Existing problem:**

A good customer chat bot should minimize involvement of customer agent to chat with a customer to clarify their doubts. So to achieve this we should include a virtual agent in chat bot so that it will take care of real involvement of customer agent and customer can clarify his doubts with fast chat bots. In general we can say that the chat bot is the platform where the user can clarify his various doubts about the product which they want to purchase and they have the full rights to know about the products. Here we have to clarify the user doubts without the help of the agent and if the chat bot can't answer they can contact the agent or any other person.

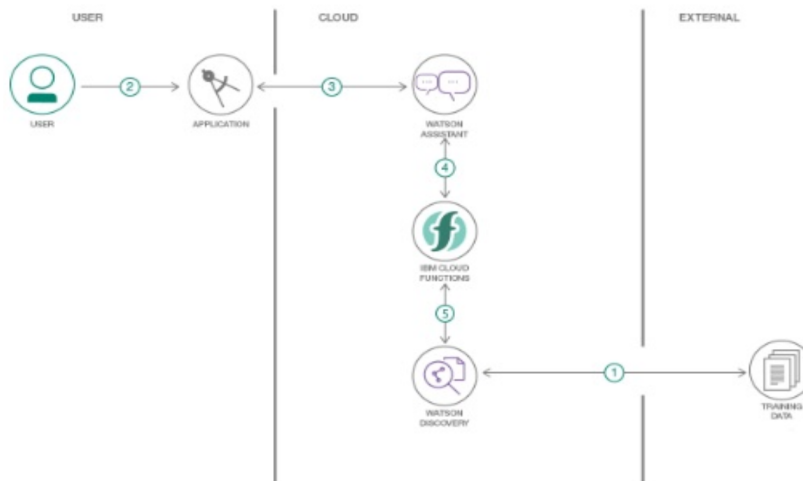
### **b.proposed solution:**

For the above problem to get solved we have to put a virtual agent in chat bot so it can understand the queries that are posted by customers. The virtual agent should be trained from some insight records based on company background so it can answer queries based on the product or related to company. In this project I have used Watson Discovery to achieve the above solution. And later including Assistant and Discovery on Node-red.

We use functions to integrate the Discovery and then Assistant and then the integration of all these are done in Node-red. The Discovery contains the manuals where we train the model and then with the help of functions we integrate it to Assistant.

### 3. Theoretical Analysis

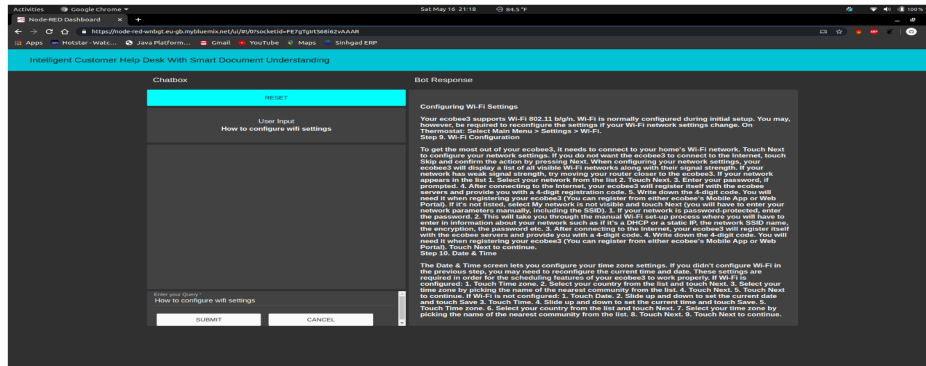
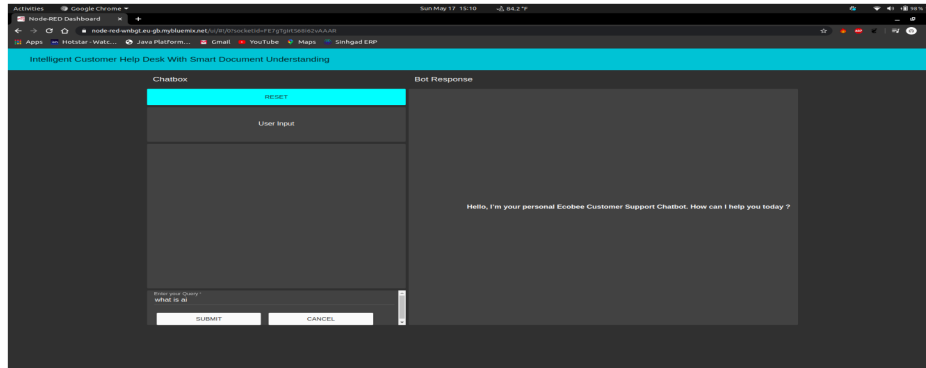
Block / Flow Diagram

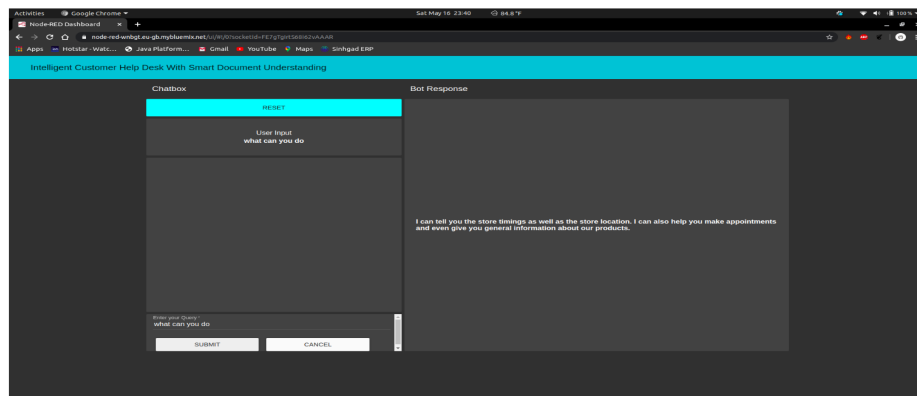
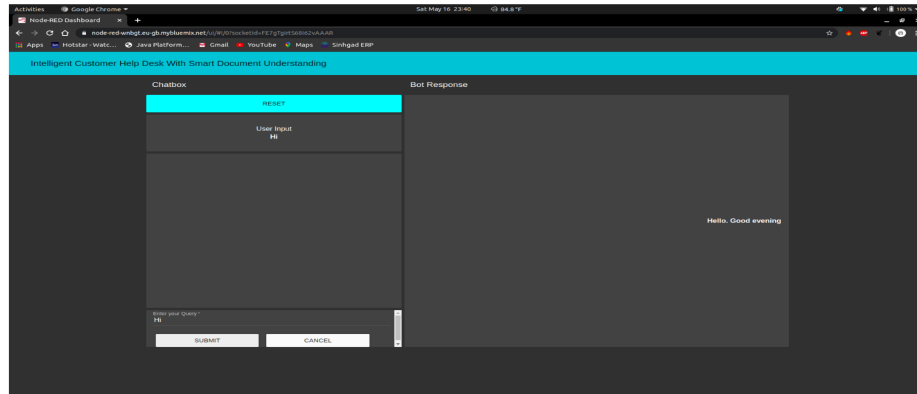


Hardware / Software Designing

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



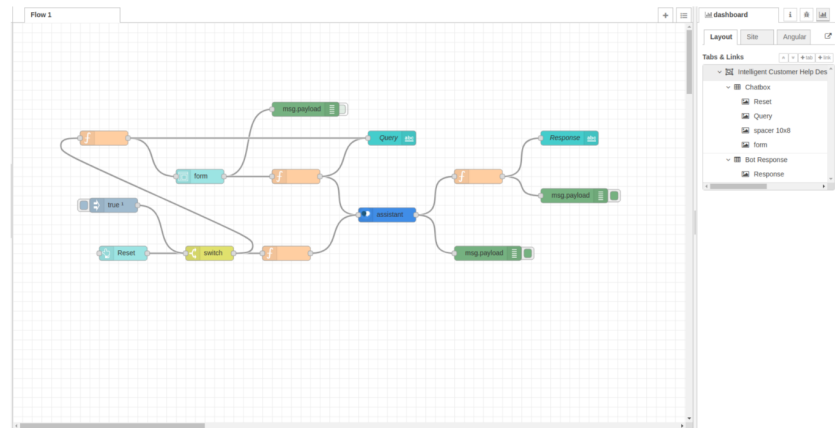




## 5. Flowchart

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

- Inject
- Debug
- ui\_Form
- ui\_Text
- ui\_Button
- Function
- Switch
- Assistant



## **6. Results**

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

URL for UI Dashboard : <https://node-red-wnbgt.eu-gb.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 cutomers 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 own 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://developer.ibm.com/articles/introduction-watson-discovery/>

## Appendix

### Source Code

#### Node-RED Flow code

```
[
  {
    "id": "e33fd96c.b1f498",
    "type": "tab",
    "label": "Flow 1",
    "disabled": false,
    "info": ""
  },
  {
    "id": "4719271e.f67278",
    "type": "debug",
    "z": "e33fd96c.b1f498",
    "name": "",
    "active": false,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "x": 610,
    "y": 180,
```

```

    "wires": []
  },
  {
    "id": "f09213bd.f0c9b",
    "type": "ui_text",
    "z": "e33fd96c.b1f498",
    "group": "d0a6c0ac.8980d",
    "order": 2,
    "width": 10,
    "height": 2,
    "name": "Query",
    "label": "User Input",
    "format": "{{msg.payload}}",
    "layout": "col-center",
    "x": 790,
    "y": 240,
    "wires": [],
    "inputLabels": [
      "asd"
    ]
  },
  {
    "id": "6783034d.ab58ec",
    "type": "watson-conversation-v1",
    "z": "e33fd96c.b1f498",
    "name": "",
    "workspaceid": "7f61646e-240e-43be-9e45-53f8eeeb2038",
    "multiuser": false,
    "context": false,
    "empty-payload": false,
    "service-endpoint": "",
    "timeout": "",
    "optout-learning": false,
    "x": 780,
    "y": 400,
    "wires": [
      [
        "c4de52e9.4afa1",
        "2a9036a7.6e21da"
      ]
    ]
  }

```

```

    ]
  },
  {
    "id": "3fac1262.28ad6e",
    "type": "ui_text",
    "z": "e33fd96c.b1f498",
    "group": "ff862856.2e9e98",
    "order": 1,
    "width": 15,
    "height": 13,
    "name": "Response",
    "label": "",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "x": 1160,
    "y": 240,
    "wires": []
  },
  {
    "id": "a4d26d94.bf4a9",
    "type": "debug",
    "z": "e33fd96c.b1f498",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "x": 1170,
    "y": 360,
    "wires": []
  },
  {
    "id": "c4de52e9.4afa1",
    "type": "function",
    "z": "e33fd96c.b1f498",
    "name": "",
    "func": "msg.payload.text=\"\\\"; \\nif(msg.payload.context.webhook_result_1)\\n{\\n
\\n  for(var i in msg.payload.context.webhook_result_1.results)\\n    {\\n

```

```

msg.payload.text=msg.payload.text+"\<br>" +msg.payload.context.webhook_result_
1.results[i].subtitle;\n
msg.payload.text=msg.payload.text+"\<br><br>" +msg.payload.context.webhook_re
sult_1.results[i].text+"\<br>"; \n  } \n  msg.payload=msg.payload.text; \n  \n} \nelse
\nmsg.payload = msg.payload.output.text[0];\nreturn msg; ",
  "outputs": 1,
  "noerr": 0,
  "x": 970,
  "y": 320,
  "wires": [
    [
      "a4d26d94.bf4a9",
      "3fac1262.28ad6e"
    ]
  ]
},
{
  "id": "7d69ebf8.9cafb4",
  "type": "ui_form",
  "z": "e33fd96c.b1f498",
  "name": "",
  "label": "",
  "group": "d0a6c0ac.8980d",
  "order": 4,
  "width": 10,
  "height": 2,
  "options": [
    {
      "label": "Enter your Query",
      "value": "query",
      "type": "text",
      "required": true,
      "rows": null
    }
  ],
  "formValue": {
    "query": ""
  },
  "payload": "",
  "submit": "Submit",

```



```

    "cancel": "Cancel",
    "topic": "",
    "x": 390,
    "y": 320,
    "wires": [
      [
        "4719271e.f67278",
        "2b6857ad.e2ef98"
      ]
    ]
  },
  {
    "id": "2b6857ad.e2ef98",
    "type": "function",
    "z": "e33fd96c.b1f498",
    "name": "",
    "func": "msg.payload = msg.payload.query\\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 590,
    "y": 320,
    "wires": [
      [
        "6783034d.ab58ec",
        "f09213bd.f0c9b"
      ]
    ]
  },
  {
    "id": "2a9036a7.6e21da",
    "type": "debug",
    "z": "e33fd96c.b1f498",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "false",
    "x": 990,
    "y": 480,

```

```

    "wires": []
  },
  {
    "id": "56cf0c53.866104",
    "type": "ui_button",
    "z": "e33fd96c.b1f498",
    "name": "",
    "group": "d0a6c0ac.8980d",
    "order": 1,
    "width": 10,
    "height": 1,
    "passthru": false,
    "label": "Reset",
    "tooltip": "",
    "color": "",
    "bgcolor": "cyan",
    "icon": "",
    "payload": "true",
    "payloadType": "bool",
    "topic": "",
    "x": 230,
    "y": 480,
    "wires": [
      [
        "a20966a1.6843c8"
      ]
    ]
  },
  {
    "id": "a20966a1.6843c8",
    "type": "switch",
    "z": "e33fd96c.b1f498",
    "name": "",
    "property": "payload",
    "propertyType": "msg",
    "rules": [
      {
        "t": "true"
      }
    ]
  },
],

```

```

    "checkall": "false",
    "repair": false,
    "outputs": 1,
    "x": 410,
    "y": 480,
    "wires": [
      [
        "e1a2b793.b65918",
        "3185b04c.79b1"
      ]
    ]
  },
  {
    "id": "e1a2b793.b65918",
    "type": "function",
    "z": "e33fd96c.b1f498",
    "name": "",
    "func": "msg.payload = \"welcome\\\"\\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 570,
    "y": 480,
    "wires": [
      [
        "6783034d.ab58ec"
      ]
    ]
  },
  {
    "id": "7454fe98.2d066",
    "type": "inject",
    "z": "e33fd96c.b1f498",
    "name": "",
    "topic": "",
    "payload": "true",
    "payloadType": "bool",
    "repeat": "",
    "crontab": "",
    "once": true,
    "onceDelay": "0.0",

```

```

"x": 210,
"y": 380,
"wires": [
  [
    "a20966a1.6843c8"
  ]
]
},
{
  "id": "3185b04c.79b1",
  "type": "function",
  "z": "e33fd96c.b1f498",
  "name": "",
  "func": "msg.payload = \"\\\"\\nreturn msg;\"",
  "outputs": 1,
  "noerr": 0,
  "x": 190,
  "y": 240,
  "wires": [
    [
      "7d69ebf8.9cafb4",
      "f09213bd.f0c9b"
    ]
  ]
},
{
  "id": "d0a6c0ac.8980d",
  "type": "ui_group",
  "z": "",
  "name": "Chatbox",
  "tab": "5e75ce40.2a426",
  "order": 1,
  "disp": true,
  "width": "10",
  "collapse": false
},
{
  "id": "ff862856.2e9e98",
  "type": "ui_group",
  "z": "",

```

```

    "name": "Bot Response",
    "tab": "5e75ce40.2a426",
    "order": 2,
    "disp": true,
    "width": "15",
    "collapse": false
  },
  {
    "id": "5e75ce40.2a426",
    "type": "ui_tab",
    "z": "",
    "name": "Intelligent Customer Help Desk With Smart Document Understanding",
    "icon": "dashboard",
    "disabled": false,
    "hidden": false
  }
]

```

## Watson Cloud Function Action Code

```

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

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);
});
```







