PROJECT REPORT

Name: ***Sabreen Shakeel (shakeelsabreen@gmail.com)***

Title: ***Intelligent Customer Help Desk With Smart Document Understanding***

Category: ***Artificial Intelligence***

Internship at *:* ***smartinternz.com@2020***

1. INTRODUCTION

1.1. Project Name:

Intelligent Customer Help Desk With Smart Document Understanding

1.2. Project Summary :

We will build a chatbot that uses various Watson Al 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**: Python, IBM Cloud, IBM Watson, Node-RED, Node JS
- **Functional Requirements**: IBM cloud
- **Technical Requirements**: Al, ML, Watson Al, Python, Node JS
- **Software Requirements**: Watson Assistant, Watson Discovery, Watson Cloud Functions, Node-RED
- **Project Schedule:** 1 month.
- **Project Deliverable:** Intelligent Chatbot with Smart Document Understanding

1.3. Scope of Work :

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

2. LITERATURE SURVEY

2.1. Proposed 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 preddetermined 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 :

Steps

1. Create IBM Cloud services
2. Configure Watson Discovery
3. Create IBM Cloud Functions action.
4. Create a Node red flow to connect all the services together
5. Configure Watson Assistant.
6. Create flow and configure node
7. Deploy and run Node Red app.

1. Create IBM Cloud services

Create the following services:

a. Watson Discovery

b. Watson Assistant

c. Node Red

2. Configure Watson Discovery

Import the document

Launch the Watson Discovery tool and create a new data collection by selecting the Upload your own data option. Give the data collection a unique name. When prompted, select and upload the redmi_note_5_pro_manual.pdf file located in the data directory of your local repo.

Annotate with SDU

Now let's apply SDU to our document to see if we can generate some better query responses. From the Discovery collection panel, click the Configure data button (located in the top right corner) to start the SDU process. The goal is to annotate all of the pages in the document so Discovery can learn what text is important, and what text can be ignored.

3. Create IBM Cloud Functions action

Now let's create the web action that will make queries against our Discovery collection. Start the IBM Cloud Functions service by selecting Create Resource from the IBM Cloud dashboard. Enter functions as the filter, then select the Functions card:

From the Functions main panel, click on the Actions tab. Then click on Create. From the Create panel, select the Create Action option.

On the Create Action panel, provide a unique Action Name, keep the default package and select the Node.js 10 runtime.

Click the Create button to create the action.

Once your action is created, click on the Code tab:

In the code editor window, cut and paste in the code from the disco-action.js file found in the action's directory of your local repository. The code is pretty

straight-forward - it simply connects to the Discovery service, makes a query against the collection, then returns the response.

If you press the Invoke button, it will fail due to credentials not being defined yet. We'll do this next. Select the Parameters tab:

Add the following keys:

- **url**
- **environment_id**
- **collection_id**
- **iam_apikey**

For values, please use the values associated with the Discovery service you created in the previous step. Now that the credentials are set, return to the Code panel and press the Invoke button again. Now you should see actual results returned from the Discovery service:

Next, go to the **Endpoints panel:**

Click the checkbox for Enable as Web Action. This will generate a public endpoint URL. Take note of the URL value, as this will be needed by Watson Assistant in a future step.

To verify you have entered the correct Discovery parameters, execute the provided curl command. If it fails, re-check your parameter values.

4. Configure Watson Assistant

Launch the Watson Assistant tool and create a new dialog skill. Select the Use sample skill option as your starting point. This dialog skill contains all of the nodes needed to have a typical call center conversation with a user.

Add new intent

The default customer care dialog does not have a way to deal with any questions involving outside resources, so we will need to add this. Create a new intent that can

detect when the user is asking about operating the Product. From the Customer Care Sample Skill panel, select the Intents tab.

Click the Create intent button. Name the intent **#Product_Information**, and at a minimum, enter the following example questions to be associated with it.

Create new dialog node

Now we need to add a node to handle our intent. Click on the Dialog tab, then click on the drop-down menu for the Small Talk node, and select the Add node below option.

Name the node **"Ask about product"** and assign it our new intent. This means that if Watson Assistant recognizes a user input such as **"how do I set the time?"**, it will direct the conversation to this node.

Enable webhook from Assistant

Set up access to our Webhook for the IBM Cloud Functions action you created in

Step #4. Select the Options tab:

Enter the **public URL endpoint** for your action. Return to the Dialog tab, and click on the Ask about product node. From the details panel for the node, click on Customize, and **enable Webhooks** for this node: Click Apply.

The dialog node should have a Return variable set automatically to **$webhook_result_1.** This is the variable name you can use to access the result from the Discovery service query.

Test in Assistant Tooling

From the Dialog panel, click the Try it button located at the top right side of the panel. Enter some user input: Note that the input "how do I turn on the heater?" has triggered our Ask about product dialog node, which is indicated by the #Product_Information response. And because we specified that $webhook_result_1.passages be the response, that value is displayed also. You can also verify that the call was successfully completed by clicking on the Manage Context button at the top right. The response from

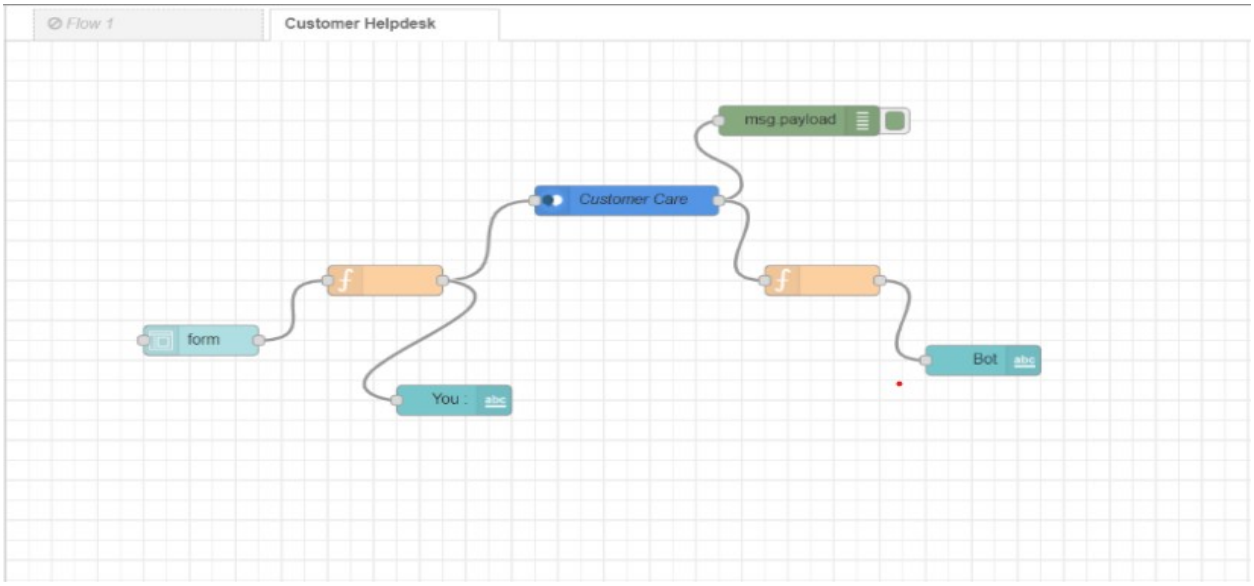
the Discovery query will be stored in the $webhook_result_1 variable.

5. Create flow and configure node:

At first go to manage palette and install dashboard.

Now, Create the flow with the help of following node:

1. ***Inject***
2. ***Assistant***
3. ***Debug***
4. ***Function***
5. ***Ui_Form***
6. ***Ui_Text***

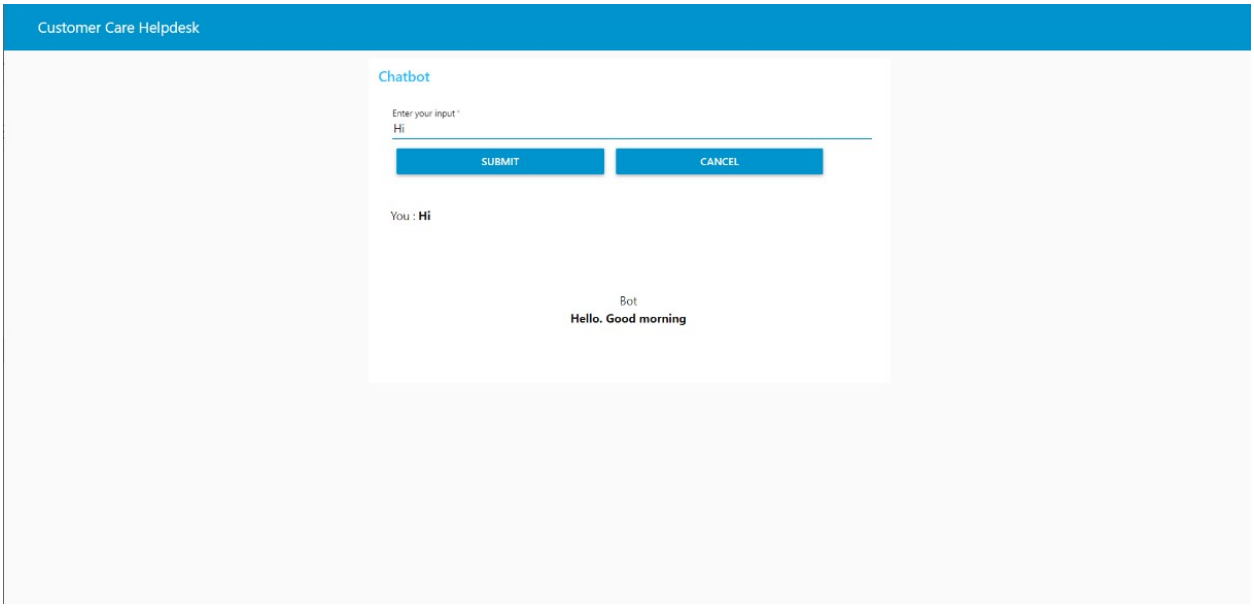


6. Deploy and run Node Red app：

Deploy the Node Red flow.

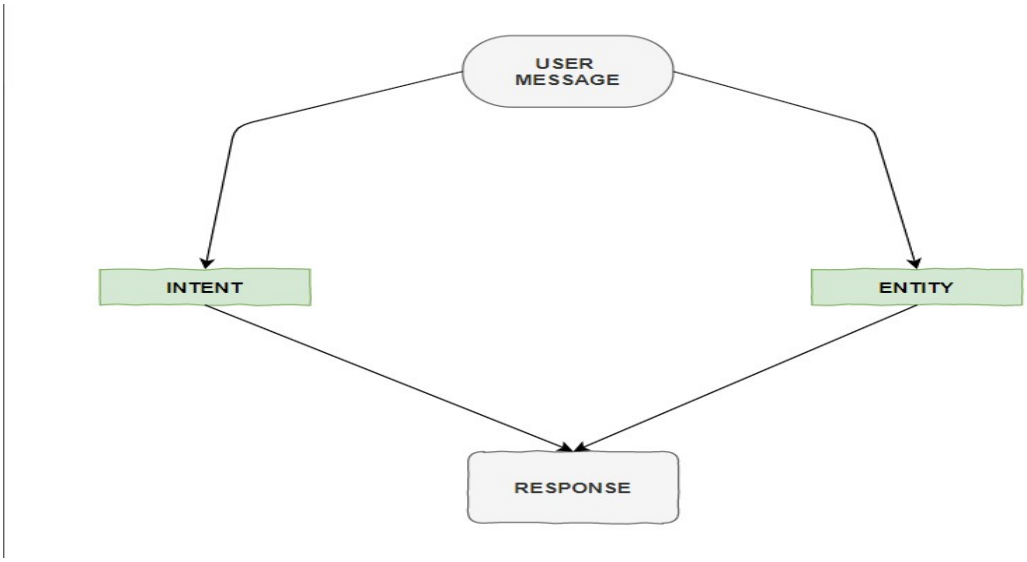
Then copy the link url upto .net/ and paste at a new tab by ui at the end of the url, like this,

https://node-red-lhbqb.eu-gb.mybluemix.net/ui



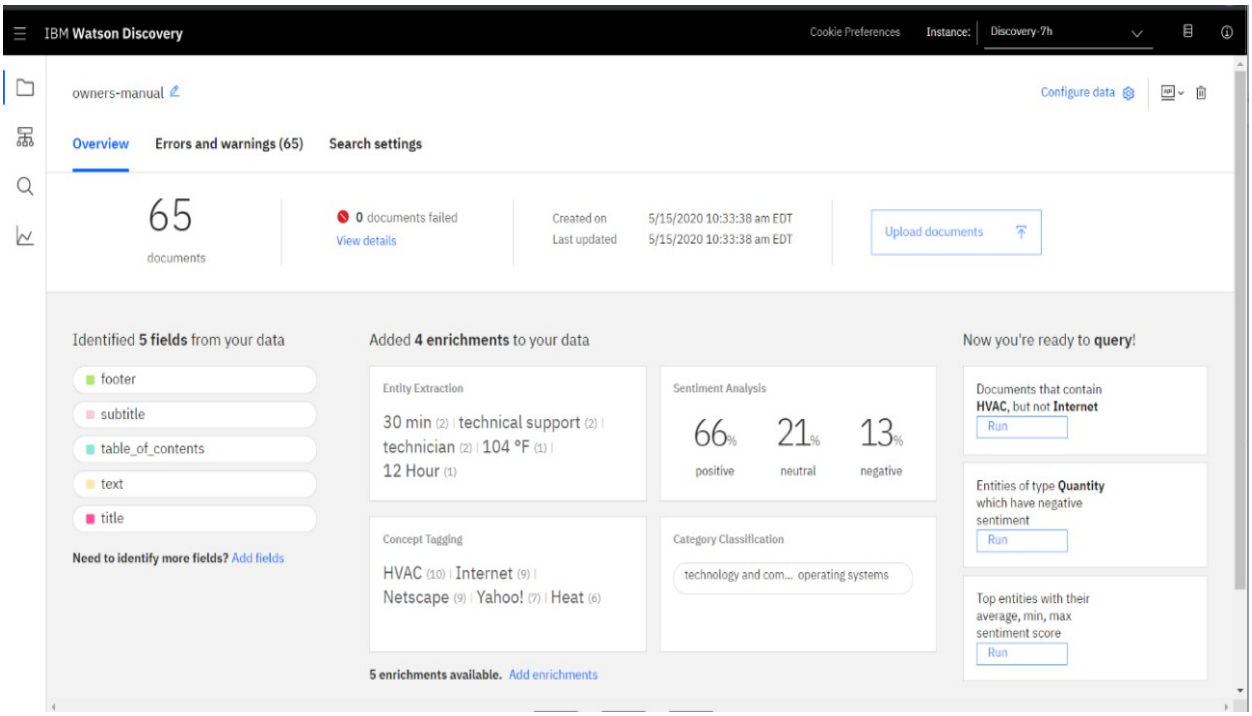
3. THEORETICAL ANALYSIS

3.1 Block Diagram :

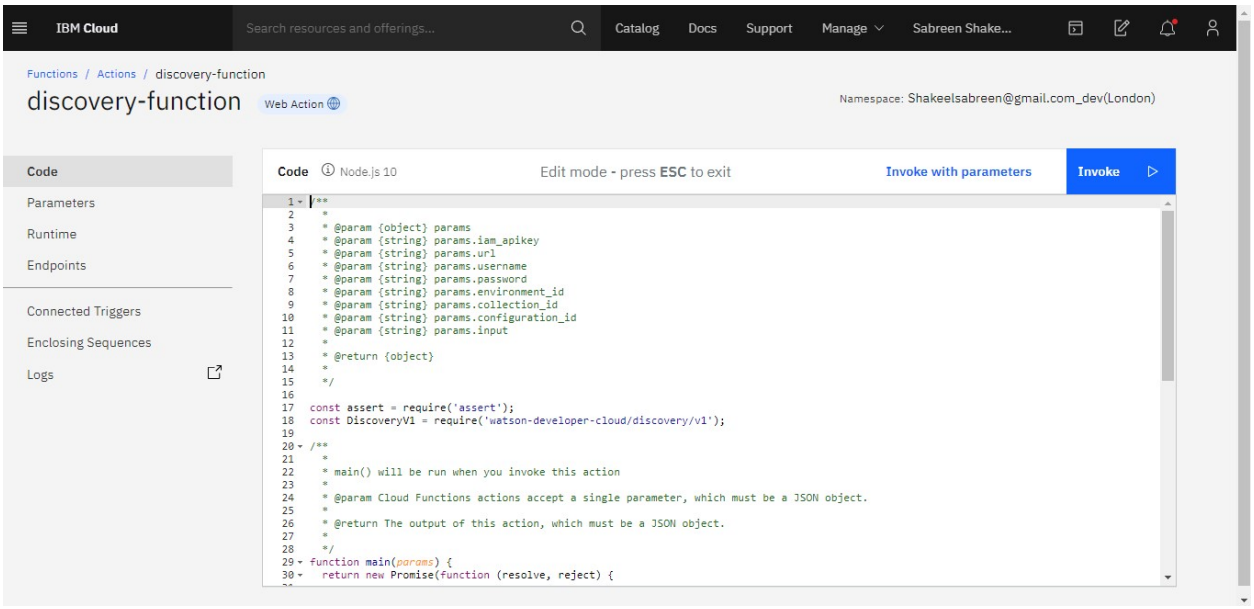


4. EXPERIMENTAL INVESTIGATIONS

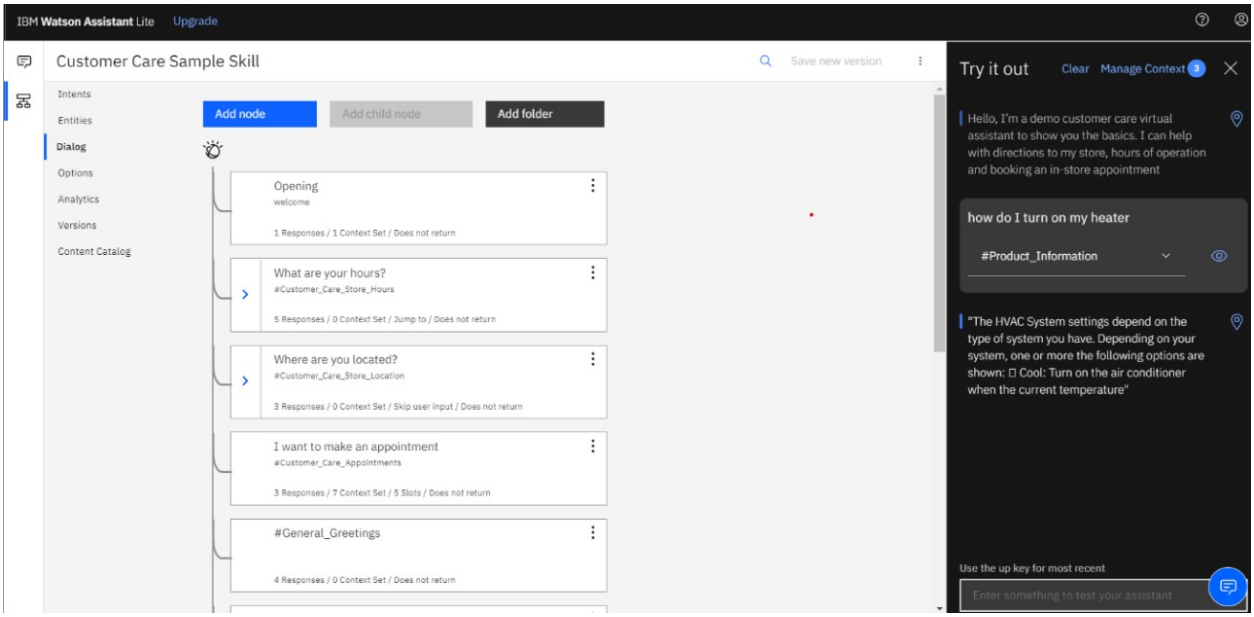
- **Watson Discovery :**



- **Cloud Function:**

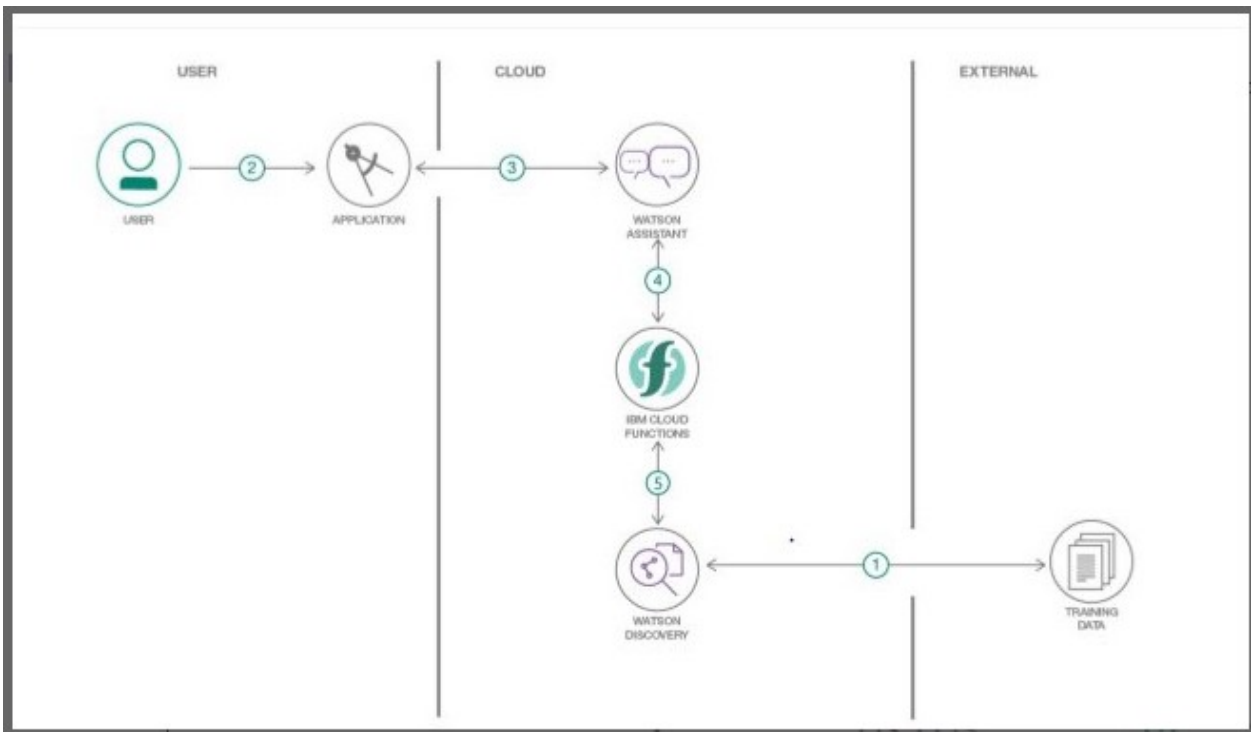


- **Watson Assistant:**



6. FLOWCHART :

This is the flow how we are going to do the tasks for the proposed problems.



1. The document is annotated using Watson Discovery SDU

2. The user interacts with the backend server via the app UI. The frontend app UI is a chatbot that engages the user in a conversation.

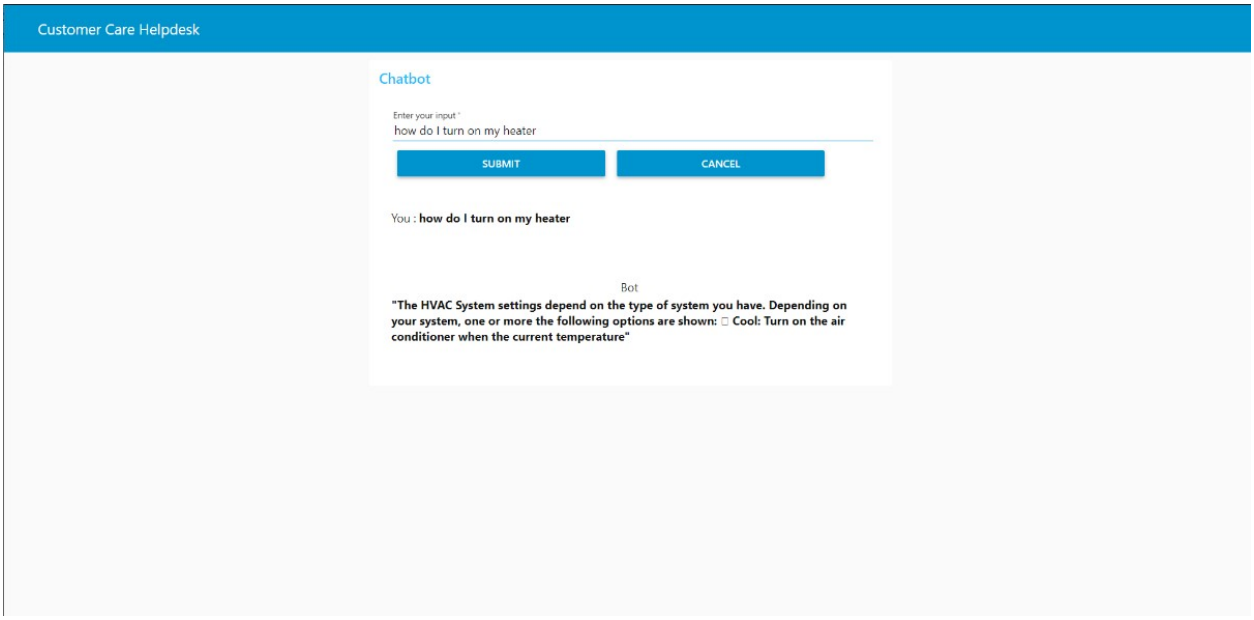
3. Dialog between the user and backend server is coordinated using a Watson Assistant dialog skill.

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

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

6. RESULTS

The chatbot was successfully made using Watson assistant and using SDU. All the services were integrated using Node Red Application.



7. ADVANTAGES & DISADVANTAGES

7.1. Advantages:

- Companies can use these to decrease the work flow to the representatives.
- Reduce the number of reps.
- Cost Efficient.
- Decrease in the number of calls diverted to representatives.
- Less work load on employees.

7.2. Disadvantages:

- Sometimes the chatbot misleads the customers.
- The discovery returns wrong results when not properly.
- Giving same answer for different sentiments.
- Sometimes is unable to connect the customer sentiments and intents.

8. APPLICATIONS

- it can be used to deploy as Customer Helpdesk for small scale products as their manual usually has the solution for the users problems.
- it can be deployed in popular social media applications like **Facebook**, **Slack** and **Telegram.**
- Chatbot can be deployed at any website to clear the basic doubts of the customer.

9. CONCLUSION

By following the above-mentioned steps, we can create a basic chatbot which can help us to answer the basic questions of the customer or user related to location of the office, working hours and the information about the product. We successfully create the intelligent help desk smart chatbot using Watson Assistant, Watson Cloud Function, Watson Discovery and Node-Red.

10. FUTURE SCOPE

We can import the pre-built node-red flow and can improve our UI, moreover we can make a database and use it to show the recent chats to the customer. We can also improve the results of discovery by enriching it with more fields and doing the Smart Data Annotation more accurately. We can get the premium version to increase the scope of our chatbot in terms of calls and requests. We can also include Watson text to audio and Speech to text services to access the chatbot hands free, These are few of the future scopes which are possible.

11. BIBLIOGRAPHY

1. https://www.ibm.com/cloud/architecture/tutorials/cognitive_discovery
2. https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started
3.https://developer.ibm.com/recipes/tutorials/how-to-create-a-watson-chatbot-on-nodered/
4.https://developer.ibm.com/recipes/tutorials/how-to-create-a-watson-chatbot-on-nodered/ odered/
5. https://github.com/IBM/watson-discovery-sdu-with-assistant
6. https://youtu.be/-yniuX-Poyw