**INTRODUCTION**

Overview

This project is to make a Customer Help Desk with Smart Socument Understanding.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.

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.

* 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

Purpose

To explore IBM Cloud platform

To explore IBM Watson Usercases

To learn about Watson Assistance

To learn about Watson Discovery

To explore IBM Cloud Functions

To get an understanding on IBM Watson,IBM assistant and NODE-RED and how to integrate them.

**LITERATURE SURVEY**

Existing problem

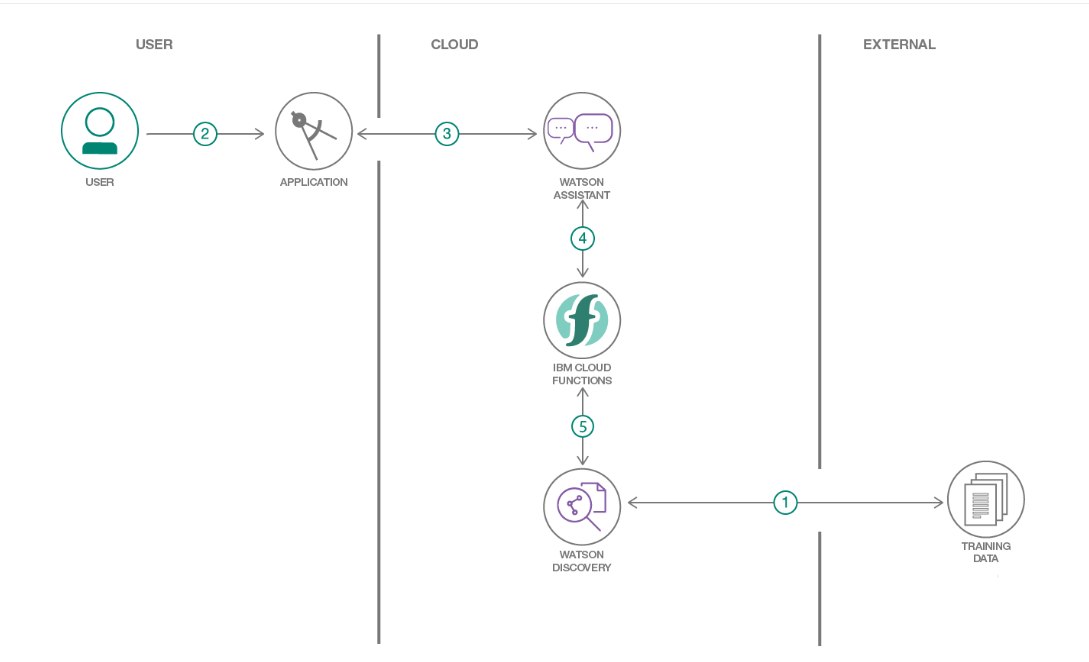
Customers facing issues as the customer support is not always available.

Proposed solution

To creating a chatbot using IBM Watson.

**THEORITICAL ANALYSIS**

Block diagram



Hardware / Software designing

1. [Create IBM Cloud services](https://github.com/IBM/watson-discovery-sdu-with-assistant/blob/master/README.md#2-create-ibm-cloud-services)

2. [Configure Watson Discovery](https://github.com/IBM/watson-discovery-sdu-with-assistant/blob/master/README.md#3-configure-watson-discovery)

3. [Create IBM Cloud Functions action](https://github.com/IBM/watson-discovery-sdu-with-assistant/blob/master/README.md#4-create-ibm-cloud-functions-action)

4. [Configure Watson Assistant](https://github.com/IBM/watson-discovery-sdu-with-assistant/blob/master/README.md#5-configure-watson-assistant)

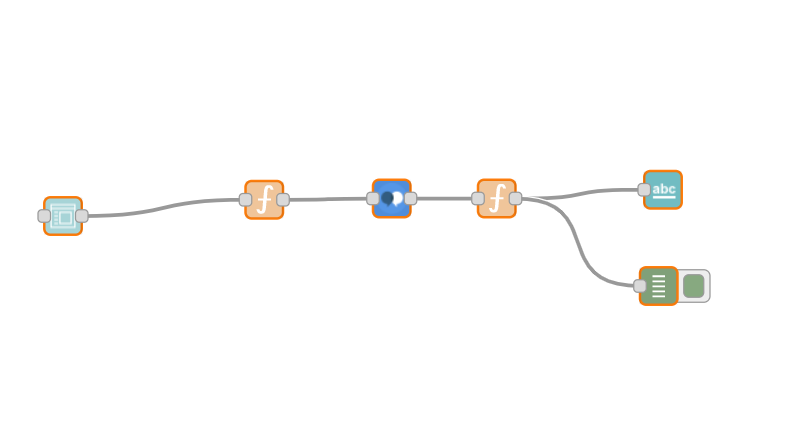
5. Create a web dashboard

6. Integrate the services using NODE-RED

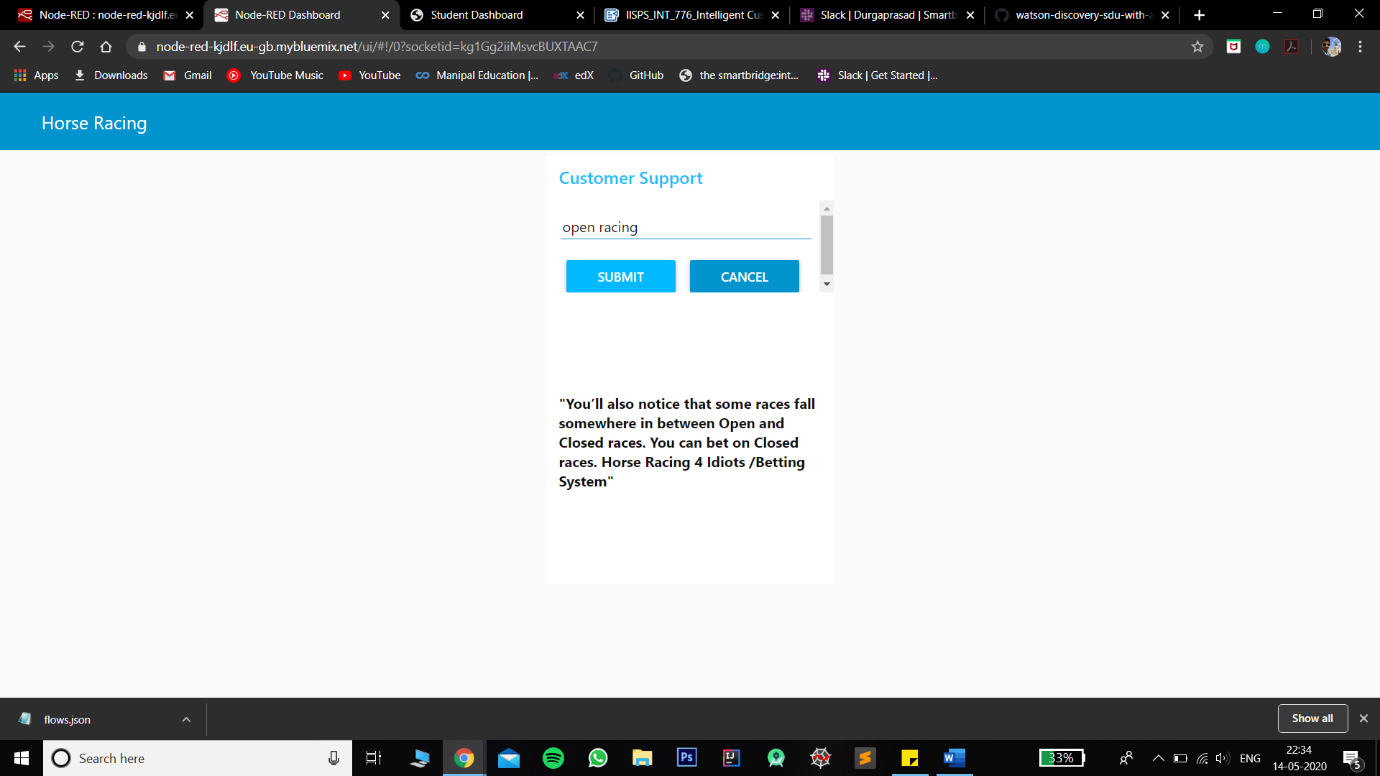
**EXPERIMENTAL INVESTIGATIONS**

It has been found that chatbots prove to be very effective in providing a simple interface for customers to interact with and have great scope of development in future.

**FLOWCHART**



**RESULT**



|  |
| --- |
| **ADVANTAGES & DISADVANTAGES**    Advantages  1. The customer does not have to wait.  2. The customer support handling gets easier.  3. Cost effecive service is provided.      Disadvantages  1. Chat bots are not intelligent enough to make decisions.  2. They are annoying and not user-friendly sometimes.  3. Fail to answer the questions out of their memory. |
| **APPLICATIONS**    Chatbots have numerous applications.They provide customer assistance.Large amounts of information is made available easily.They can be used to book cabs,flights,get product recommendations,etc.    **CONCLUSION**    Chatbots are an intelligent way to provide customer service on finger tips which can be used effectively to improve the customer experience.    **FUTURE SCOPE**    -They can be used to provide medical assistance.  -They can be used to give the latest information about stocks to the user.  -They can be used to buy things online.  -They can be used by banks to make funds transfer easier.          **BIBILOGRAPHY**  www.searchenginejournal.com  www.cxservice360.com  www.ecommerce-nation.com    **APPENDIX**  Source code  funtion nodes:  msg.payload=msg.payload.input;  return msg;    msg.payload=msg.payload.output.text[0]  return msg;  cloud function  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);  });  });  }  project demo:  <https://youtu.be/9aoOnr93cs0> |
|  |