Project Report

**Name:** Yash Wani

**Application ID:** SPS\_APL\_20200003530

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

**Category**: *Artificial Intelligence*

*Internship at smartinternz.com*

1. **Introduction**

**1.1 Project Overview:**

A simple customer care chatbot can answer simple questions, such as store locations and hours, directions, and maybe even making appointments. But when a question comes out of the scope, the bot asks to rephrase the question, gives an error or asks to talk to the real person.

In this project, there will be another option where if a question is asked out of the scope it will be answered by the Watson assistant.

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.

**Project Requirements**: Python, IBM Cloud, IBM Watson

**Functional Requirements**: IBM cloud

**Technical Requirements**: AI, ML, WATSON AI, PYTHON

**Software Requirements**: Watson assistant, Watson discovery.

**1.2 Purpose**

* Create a customer care dialog skill in Watson Assistant.
* Upload a manual in Watson Discovery and 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.

1. **Literature Survey**
   1. **Existing Problem**

A simple customer care chatbot can answer only typical questions such as store timings, Greetings, Location etc. But whenever the question comes out of the scope, the chatbot asks us to rephrase the question or asks to talk to a real person.

* 1. **Proposed Solution**

# Steps

* 1. Create IBM Cloud account.
  2. Import a manual to the Watson Discovery.
  3. Configure Watson Discovery.
  4. Create IBM Cloud Function action.
  5. Configure Watson Assistant.
  6. Create a Node Red Flow to connect all the services.
  7. Create flow and configure nodes.
  8. Deploy and run Node Red app.

# 1. Create IBM Cloud Account

1. Create an IBM account by logging in into https://cloud.ibm.com/ and clicking on Create an IBM account.
2. Fill the details and create an IBM account
3. Verify your email id and then login into IBM account by clicking on Login.

# 2. Import a manual to the Watson Discovery

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 HP Manual file located in the data directory of your local repository.

**3. Configure Watson Discovery**

Now let's apply Smart Document Understanding(SDU) to our document. 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.

**4. Create IBM Cloud Function 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 simply connects the Discovery service to the cloud function, makes a query against the collection, then returns the response.

Next, select the Parameters tab:

Add the following keys:

* url
* environment\_id
* collection\_id
* iam\_apikey
* Configuration ID

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.

To verify you have entered the correct Discovery parameters, execute the provided url command.

# 5. 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 be adding a new intent. 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 enter some examples to be asked.

*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.

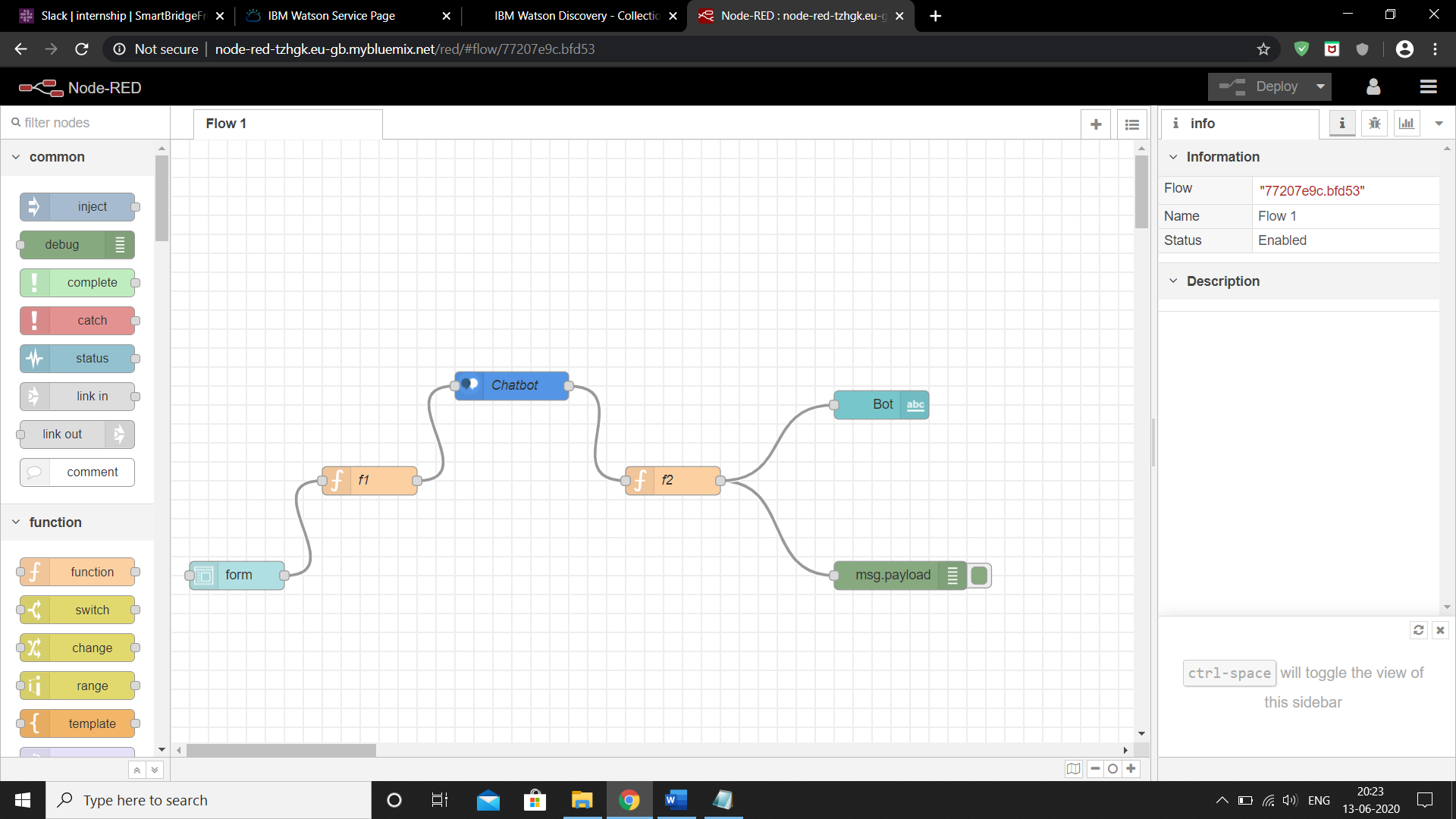
Test in Assistant Tooling

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.

# 6. Create flow and configure node:

At first go to manage palette and install dashboard. Now, Create the flow with the help of following node:

* Inject
* Assistant
* Debug
* Function
* Ui\_Form
* Ui\_Text

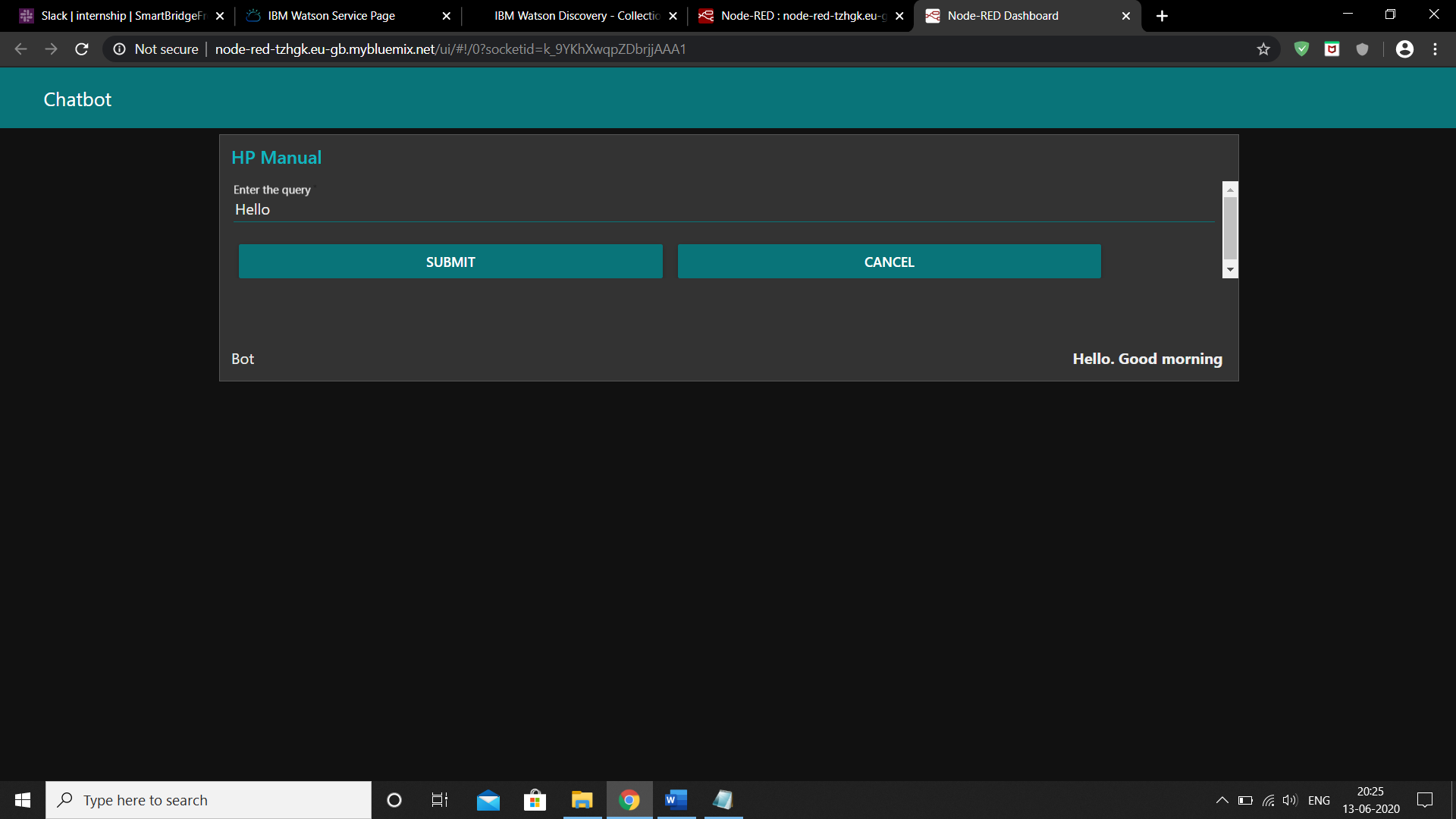


**7. Deploy and run Node Red app.**

Deploy the Node Red flow.

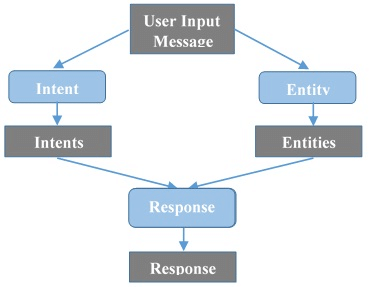
Then click on the dashboard option in the top right corner of the screen and click the button below it. This will take you to the node red ui.

<http://node-red-tzhgk.eu-gb.mybluemix.net/ui/#!/0?socketid=k_9YKhXwqpZDbrjjAAA1>



**3. Theoretical Analysis**

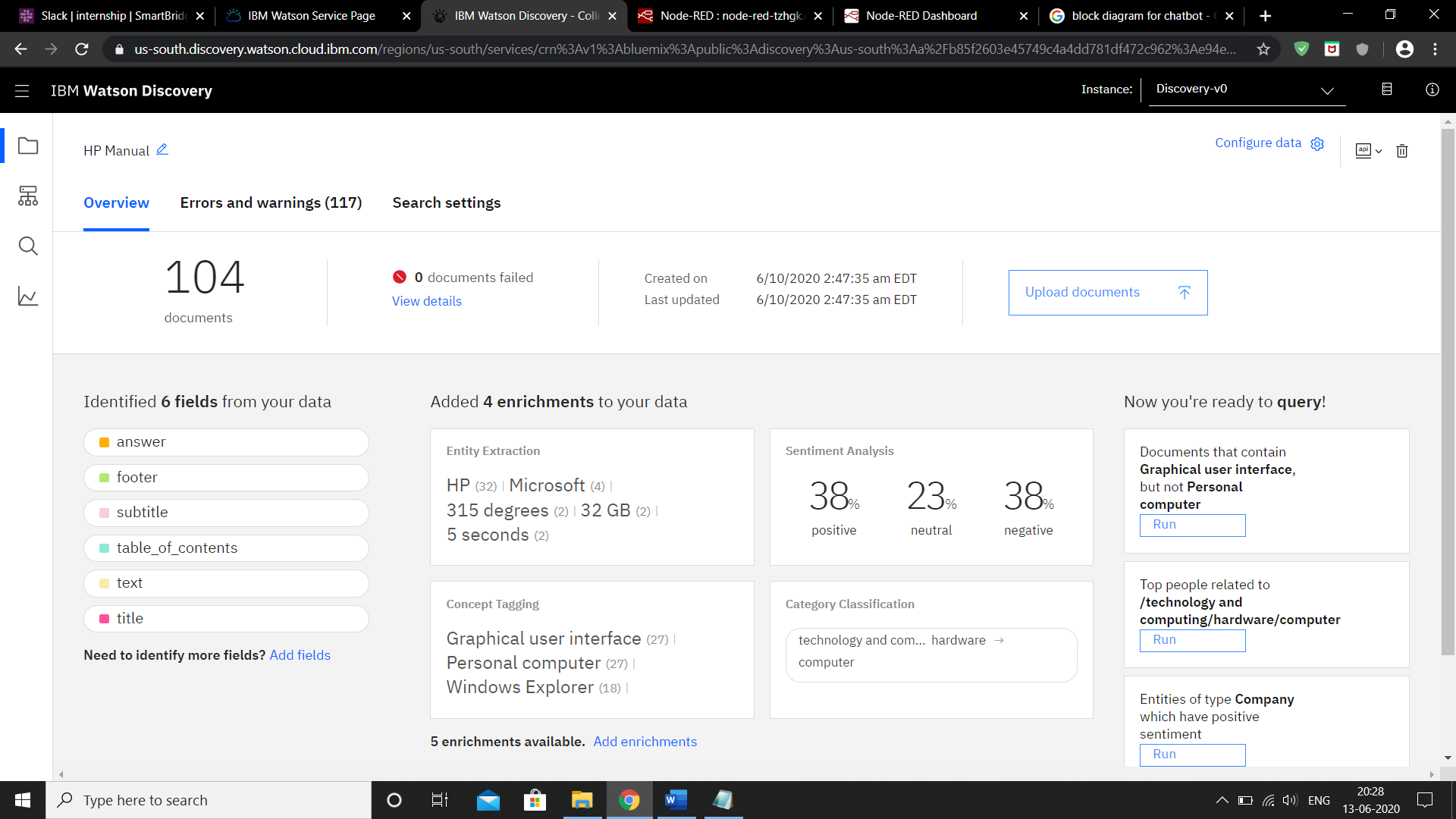
**3.1 Block Diagram**



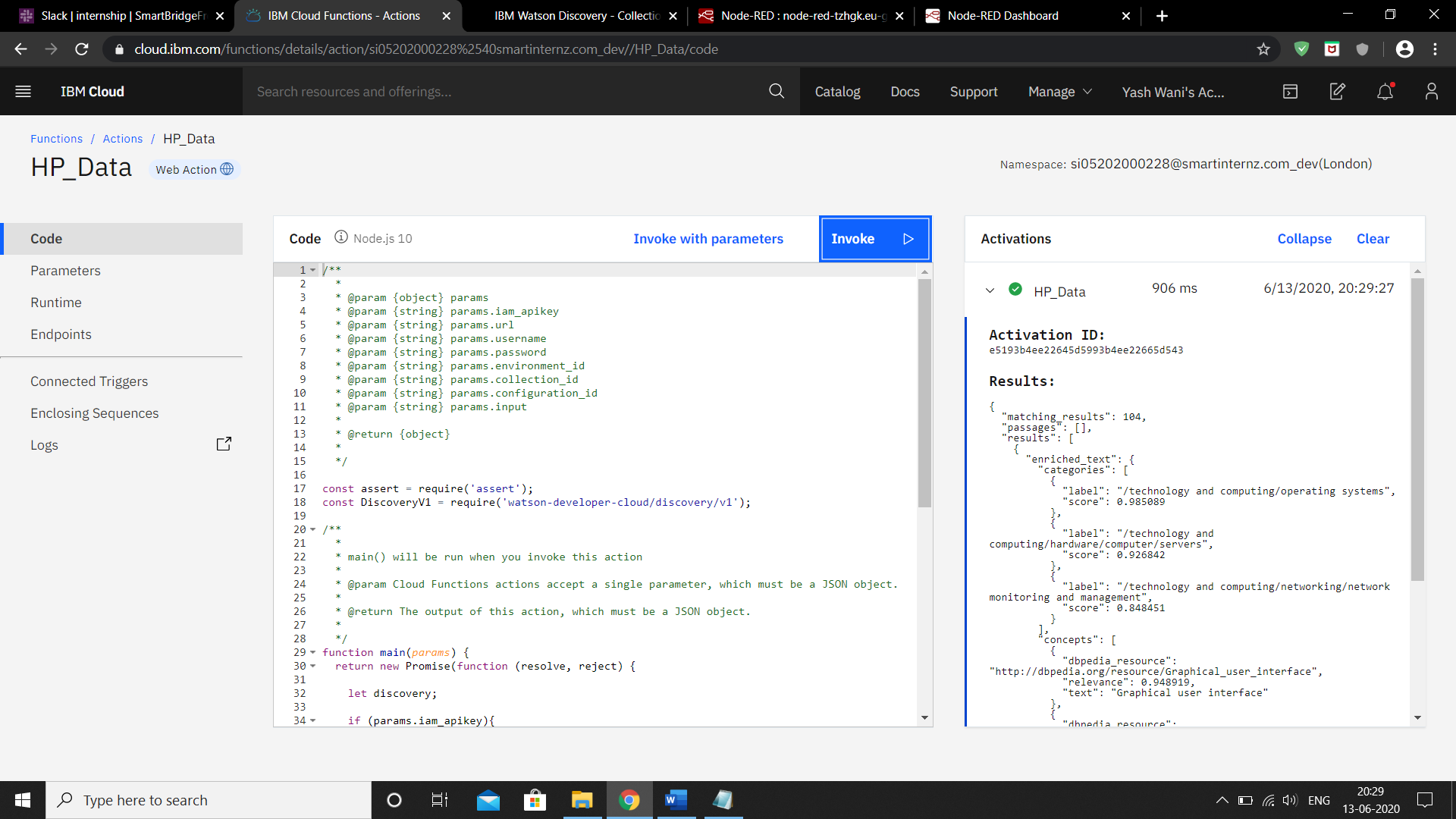
**3.2 Hardware and Software Designing**

* **Project Requirements**: Python, IBM Cloud, IBM Watson
* **Functional Requirements**: IBM cloud
* **Technical Requirements**: AI, ML, WATSON AI, PYTHON
* **Software Requirements**: Watson assistant, Watson discovery.

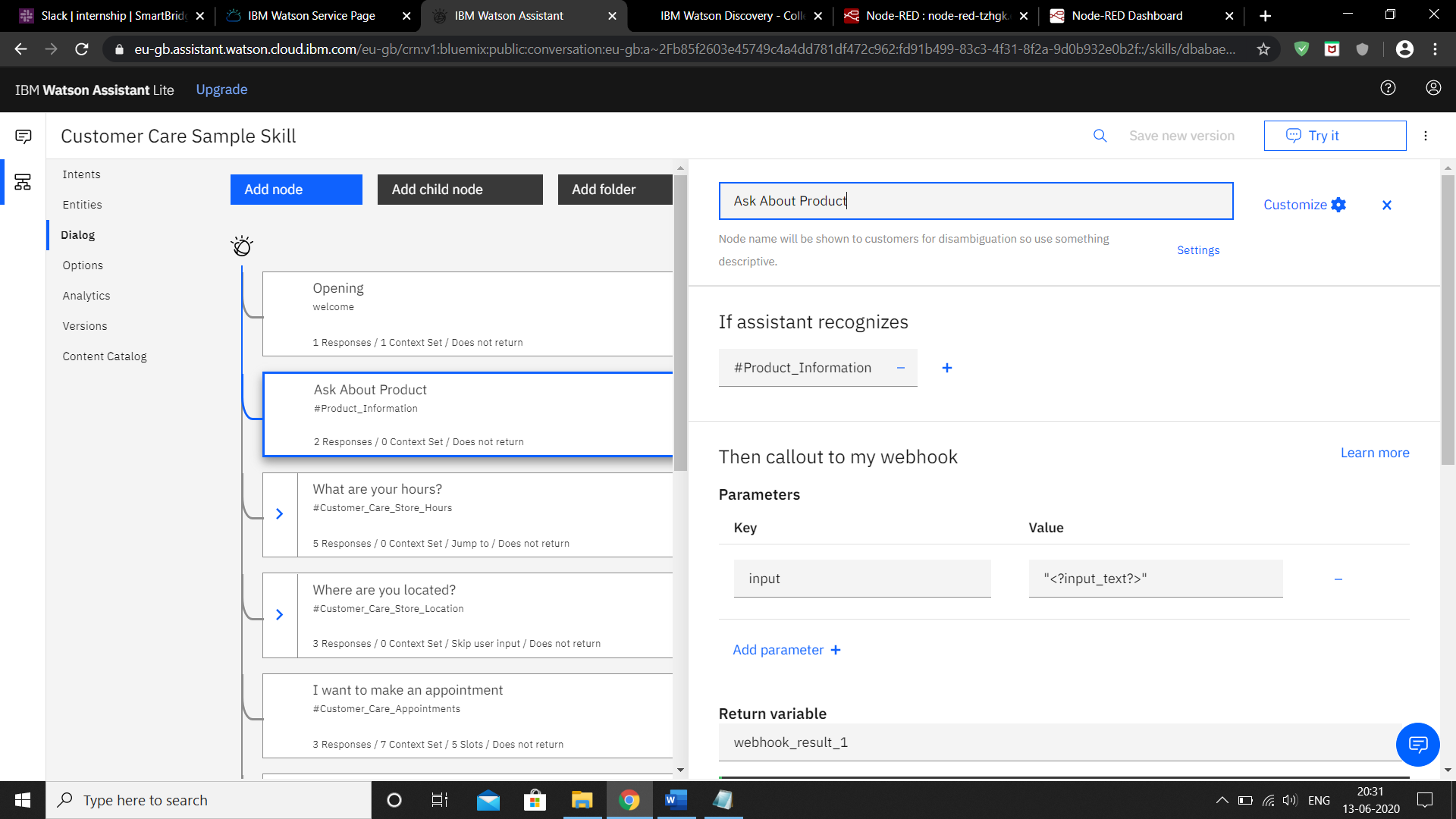
**4. Experimental Investigations**



**Watson Discovery**



**Cloud Function**



**Watson Assistant**

1. **Flowchart**

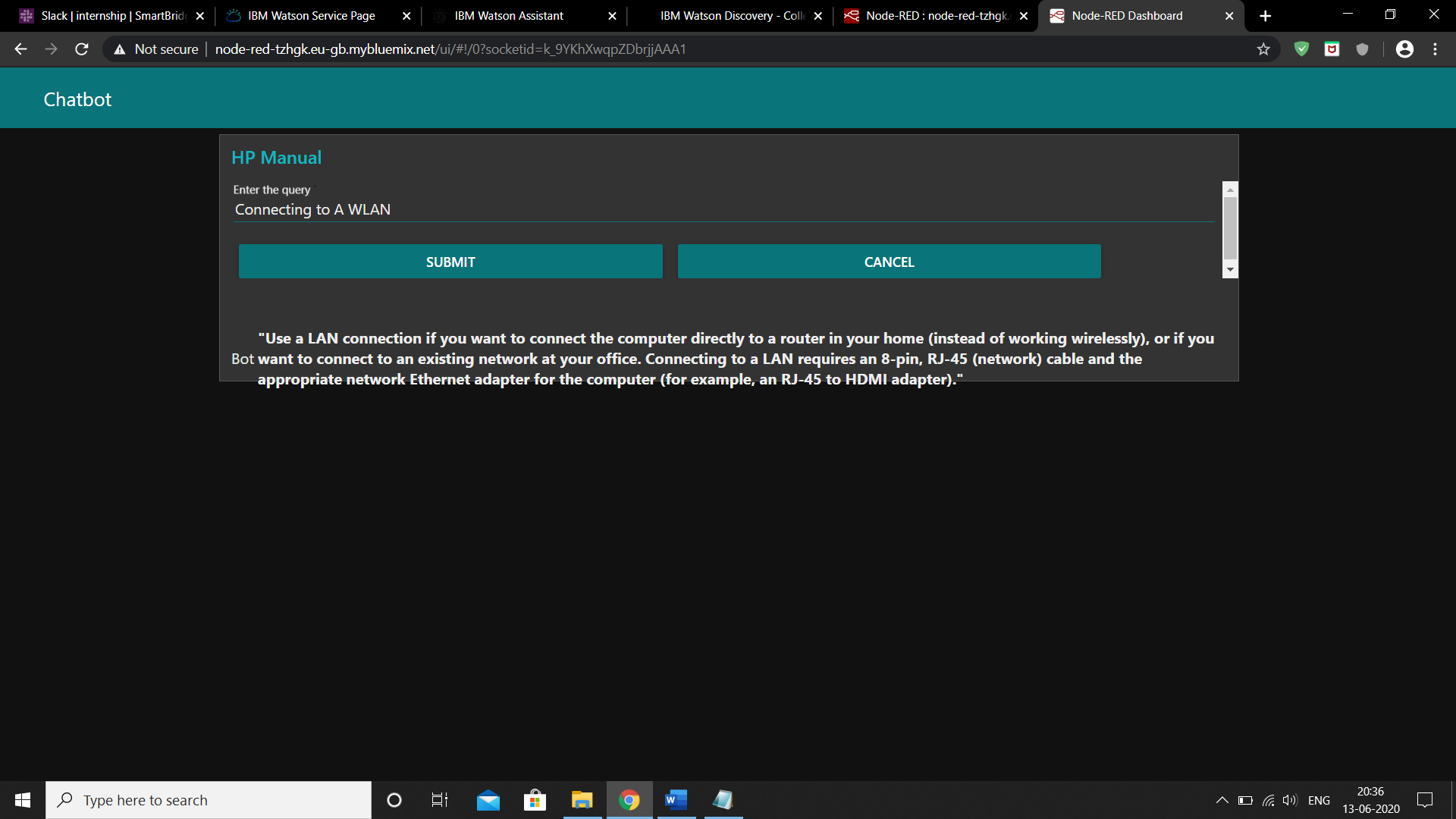
This is the flowchart according to which we will be carrying out our project.

The document is annotated using Watson Discovery SDU

* 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.
* Dialog between the user and backend server is coordinated using a Watson Assistant dialog skill.
* If the user asks a product operation question, a search query is passed to a predefined IBM Cloud Functions action.
* 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 and Disadvantages**

**Advantages:**

1. Reduced costs: Chatbots eliminate the need for labor during online interaction with customers. This is obviously a great advantage for companies that receive multiple queries at once.
2. 24/7 Availability: Unlike humans, once we install a chatbot, it can handle queries at any time of day. Thus, the customer does not have to wait for a commercial of the company to help him. This also allows companies to monitor customer « traffic » during non-working hours and contact them later.
3. Learning and updating: AI-based chatbots are able to learn from interactions and update independently. This is one of the main advantages. When you hire a new employee, you have to train them continuously. However, chatbots « form » themselves (with certain limitations, of course).
4. Management of multiple clients: Humans can serve a limited number of customers at the same time. This restriction does not exist for chatbots, and they can manage all the necessary queries simultaneously. This is one of the main advantages of using chatbot, as no customer is left unattended and you are solving different problems at the same time. There are chatbots companies already working on developing voice chatbot services.

**Disadvantages:**

This definition however often leads to two potential misconceptions.

1. The biggest misconception that arises is that a chatbot is a bot that converses with a human in the way that another human would converse with a human. Software or even a robot (the digital part of the robot is of course software) that communicates with a human in natural language is not difficult to imagine. Science fiction is full of examples.

2. The second misconception is that a chatbot communicates using only text or voice. Actually chatbots allow users to interact with them via graphical interfaces or graphical widgets, and the trend is in this direction. Many chat platforms including WeChat, Facebook Messenger and Kik allow web views on which developers can create a completely customized graphical interfaces.

**8. Applications**

* Content delivery: Media Publishers have realized that chatbots are a powerful way to engage with their audiences and monitor engagement to gain valuable insights on reader interests.
* Order Food: Various fast food giants like KFC and Pizza Hut have invested in Chatbots that enable customers to place their orders through conversations. Taco Bell went a step further to improve the conversational experience by giving their Chatbot named TacoBot some personality.
* Book Flights: Icelandair’s chatbot gives their customers the ability to search for and book flights in a text-based conversational manner. Instead of drop-down menus, customers enter the information themselves.
* Companionship: Russian technology company Endurance developed its companion chatbot for Senior People and Patients with Alzheimer’s Disease. 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. The chatbot asks questions, reacts to the answers, is able to speak on various topics, and share interesting news and facts from Google.
* Health Care: Chatbots have also made their way into health care by easing the burden on medical professionals by facilitating faster medical diagnosis, answering health-related questions, booking appointments and lots more. A Chatbot like Super Izzy can track menstrual cycles, dates and fertile windows.

1. **Conclusion**

This chatbot will be useful for the user to ask the assistant the queries related to the Product and will give them clear guidance about the product. If the Assistant doesn’t know about a certain query, it will redirect to the correct person for it. Chatbots are quickly making transformational changes and allowing businesses to thrive through customer interactions. The feedback and survey through chatbots strengthen the position of businesses as they analyze the reason behind different levels of customer approval. Use of conversational AI chatbots only means better engagement and relentless need for customer satisfaction in the near future.

1. **Future Scope**

1] Smarter Virtual Assistants: Much of what virtual assistants do now are basic skills, such as retrieving data and basic computation. As natural language processing (NLP) continues to mature, virtual assistants will improve their comprehension and response capabilities, allowing for their use to become more widespread and complex. Also, as machine learning progresses, we may see virtual assistants become smarter and begin to learn and predict customer needs.

2] Integration with IoT Devices: Car speakers, smart home devices, and wearables are just a few examples where the virtual assistant is departing from its original hardware and making its way to in-context devices. These integrations ensure that virtual assistants can always be near their human and ready to support any need. It is expected that these integrations will continue at an accelerated pace throughout 2018.

3] Voice-control: Voice recognition can be added with the virtual assistant. Then the customer can control application by using his voice. Soon, we could be joining meetings with a voice command, instead of dialing in the long meeting ID and password.

4] The clearest use of chatbots right now is in customer service and online ordering, where it can automate (and in some cases solve) customer issues or complete orders without human interaction.

• Adding Natural Language Processing in the Bot to understand the User Statements.

• Adding Sentiment Analysis to predict User Sentiment during the Chat.

• Use Voice Capabilities of the Bot.

• Use Voice Recognition with Bot.

1. **Bibliography**
2. https://[www.ibm.com/cloud/architecture/tutorials/cognitive\_discovery](http://www.ibm.com/cloud/architecture/tutorials/cognitive_discovery)
3. https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started
4. https://developer.ibm.com/recipes/tutorials/how-to-create-a-watson-chatbot-on-nodered/
5. <http://www.iotgyan.com/learning-resource/integration-of-watson-assistant-to-node-red>
6. https://www.uio.no/studier/emner/matnat/ifi/IN5480/h18/deliverables/group-assignment/final-reports/finalreport-vildehos\_annassc\_martrim.pdf
7. https://github.com/IBM/watson-discovery-sdu-with-assistant
8. https://[www.youtube.com/watch?v=Jpr3wVH3FVA](http://www.youtube.com/watch?v=Jpr3wVH3FVA)
9. https://www.cxservice360.com/2018/06/27/10-interesting-applications-of-chatbots-2/

**12. YouTube Video Links**

* Project Demo - https://youtu.be/1cXMwNsv-to
* Testimonial Video - https://youtu.be/4813SAkmHKo