SVKM's NMIMS

Mukesh Patel School of Technology Management & Engineering

A REPORT ON HEALTHCARE CHATBOT

Faculty mentor:

MR HARSHAL KOTWAL

(Assistant Professor)

MPSTME, NMIMS

Shirpur Campus

Submitted By:

ABHISHEK JAIN

SAP ID: 70471117039

Course: MBA TECH CS

Batch: 2017-2022

A report submitted in partial fulfilment of the requirements of 5 years Integrated MBA (Tech) Program of Mukesh Patel School of Technology Management & Engineering, NMIMS

ACKNOWLEDGEMENT

I would like to thank my institution SVKM’s NMIMS Mukesh Patel School of Technology Management and Engineering, **Dr R. S. Gaud**, Director (SVKM’s NMIMS, Shirpur Campus), **Dr Nikhilesh Kumar Sharma**, Director (Engineering Program, MPSTME NMIMS Shirpur Campus), **Dr Narayan Chandak,** (MPSTME NMIMS, Shirpur Campus)Associate Dean **Dr Radhakrishna Rambola,** Head of Department (MPSTME NMIMS, Shirpur Campus) for introducing me to this Project, which gave me the great opportunity to expose myself to the real world besides giving me a platform to have hands-on experience to learn new technologies and apply them.

My faculty mentor **Mr. HARSHAL KOTWAL,** Assistant Professor **(**MPSTME, NMIMS, Shirpur Campus**)** has given me constant support, always motivate me to step forward and besides being a source of inspiration through her words of encouragement. Her continuous supervision and much needed guidance has really helped me in conceptualizing my project and executing the same.

I place a deep sense of gratitude towards my family members and my friends who have been a constant source of inspiration to make this project executable.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, to attain desired career objectives.

Abhishek Jain

Date: 6-July-2020

TABLE OF ILLUSTRATION

ABSTRACT

Normally Users are not aware of all the treatment or symptoms regarding the particular disease. For small problems, the user has to go personally to the hospital for a check-up which is more time-consuming. Also handling the telephonic calls for the complaints is quite hectic. Such a problem can be solved by using medical healthcare ChatBot by giving proper guidance regarding healthy living. Today's people are more likely to be addicted to the internet but they are not concerned about their personal health. Big disease can start from small problems such as headaches which feels normal but it may beginning of big diseases such as brain tumor .most of the disease can be identified by common symptoms so the disease can be predicted if the patient's body is analyzed and reported periodically.

INTRODUCTION

About the project

The purpose of this project to provide the admin has to collect the patient's medical history of records and filter it appropriately by applying data preprocessing techniques. Admin's functionalities are to collecting the appropriate medical records of the patients, handle missing values, handling categorical values, creating sparse matrix representation, feeding data to the autonomous pipeline for predictions, selecting and training an appropriate machine learning algorithm.
The visitor can perform the basic task of the visitor is to access the Chatbot from the front end and reply to its queries with a binary response (Yes/No). The first step is to input the basic details of the visitor and according to that the disease can be identified. The user just needs to enter the symptom he/she is facing and the chatbot will answer to the symptom provided accordingly. If one symptom is not enough for the prognosis, the chatbot agent will ask the follow-up symptom for a better prognosis.
The purpose of this app is to simplify the Healthcare needs:

- people using it will be able to better understand their health problems.
- It also has a text-to-speech and speech-to-text engine for a better accessibility.
- User can also ask some random funny questions to the chatbot which in response will give some witty answers to the user.
- If the user is not clear with the symptoms, he/she will be contacted by the customer care department because of the automatic email functionality.
- Deployed on multiple platforms for easy availability and accessibility.

Requirement Analysis

Requirement Analysis is a software engineering task that bridges the gap between system-level software allocation and software design. It provides the system engineer to specify software function and performance indicate software's interface with the other system elements and establish constraints that software must meet.

The basic aim of this stage is to obtain a clear picture of the needs and requirements of the end-user and also the organization. The analysis involves interaction between the clients and the analysis. Usually, analysts research a problem from any questions asked and reading existing documents. The analysts have to uncover the real needs of the user even if they don't know them clearly. During the analysis, it is essential that a complete and consistent set of specifications emerge for the system. Here it is essential to resolve the contradictions that could emerge from information got from various parties. This is essential to ensure that the final specifications are consistent.

It may be divided into 5 areas of effort.

● Problem recognition
● Evaluation and synthesis
● Modeling
● Specification
● Review

Each Requirement analysis method has a unique point of view. However, all analysis methods are related by a set of operational principles. They are:

● The information domain of the problem must be represented and understood.

● The functions that the software is to perform must be defined.

● The behaviour of the software as a consequence of external events must be defined.

● The models that depict information function and behaviour must be partitioned in a hierarchical or layered fashion.

● The analysis process must move from essential information to implementation detail.

Software Requirements Specification

Software Requirements Specification plays an important role in creating quality software solutions. The specification is basically a representation process. Requirements are represented in a manner that ultimately leads to successful software implementation. Requirements may be specified in a variety of ways. However, there are some guidelines worth following: -

● Representation format and content should be relevant to the problem.

● Information contained within the specification should be nested.

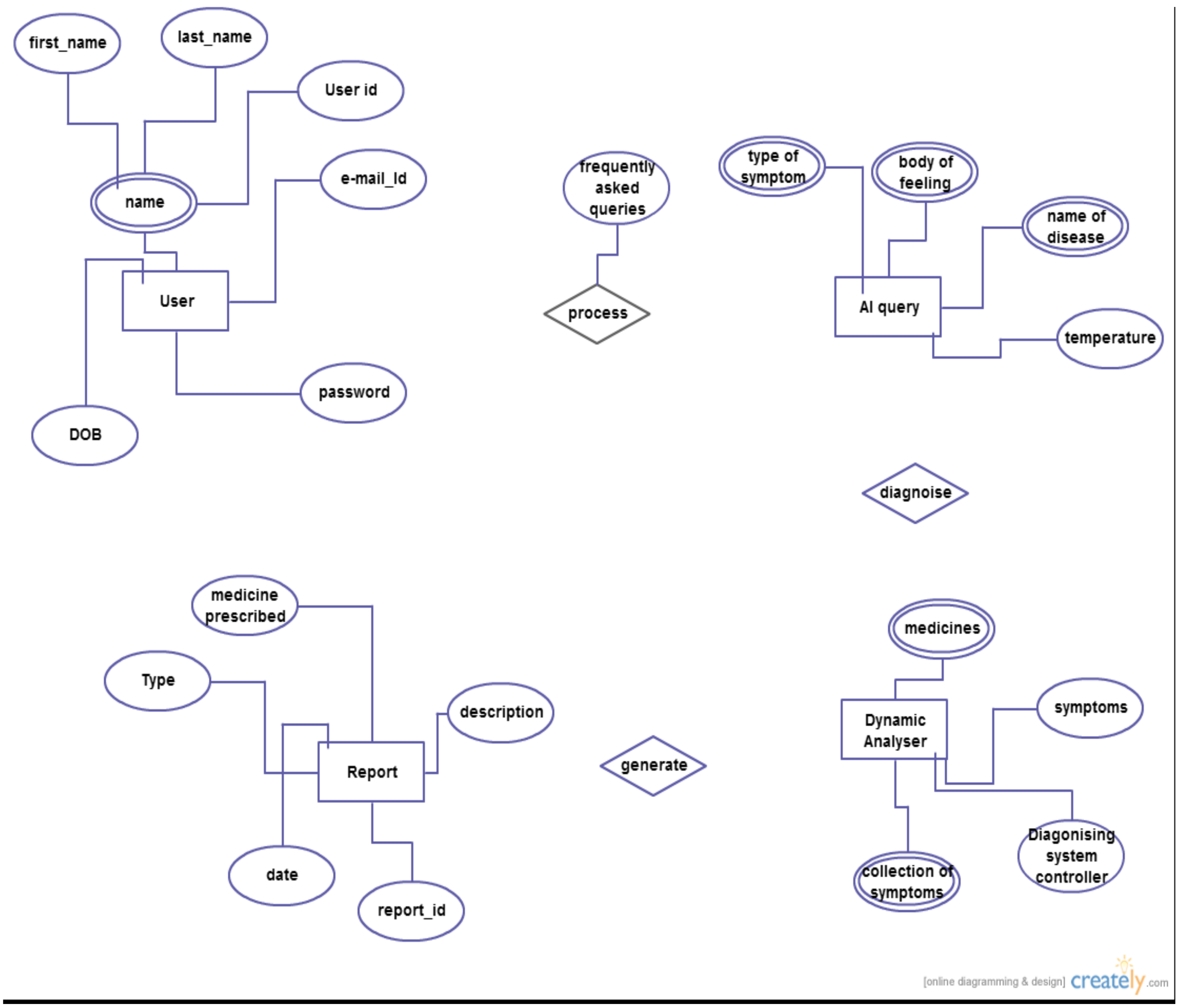
● Diagrams and other notational forms should be restricted in number and consistent in use.

● Representations should be revisable.

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of system engineering are refined by establishing a complete information description, a detailed functional and behavioural description, and indication of performance requirements and design constraints, appropriate validation. Criteria and other data pertinent to requirements.

SYSTEM DESIGN

System design is the most creative phase of system development. The term describes a final system and the process by which it is developed. The question in system design is: How the problem is to be solved?
A systematic method has to achieve beneficial results in the end. It involves starting with a vague idea and developing it into a series of steps. The series of steps for successful system design are:
The first step is to study the problem completely because we should know the goal. We should see what kind of output we require and what king of input we give so that we can get the desired result.
We should see what kind of program should be developed to reach the final goal. Then we write individual programs, which later on joining solve the specified problem. Then we test these programs and make necessary corrections to achieve the target of the programs.



IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into the working system and is giving confidence to the new system for the users i.e. will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of method to achieve the changeover, an evaluation, of change over methods. A part of planning a major task of preparing the implementation is the education of users. The more complex system is implemented, the more involved will be the system analysis and design effort required just for implementation. An implementation coordinating committee based on policies of the individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions may regarding the equipment that has to be acquired to implement the new system.

Implementation is the final and important phase. The most critical stage is in achieving a successful new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or the inability to handle certain types of transaction while using the new system.

The major elements of the implementation plan are test plan, training plan, equipment installation plan, and a conversion plan.

DETAILED OUTLINE

Healthcare chatbot will be developed using Google Dialogflow which is a part of Google cloud platform, and the backend will be implemented using python. In python we will be using Scikit learn for Data preprocessing, training, and testing. Scikit learn is a library for python programming language which includes various classification, clustering, and regression algorithms.
The second python library which we will be using is numpy, numpy is a library for the python programming language, adding support for large, multidimensional arrays and matrices along with a large collection of high level mathematical functions to operate on these arrays.
Google Dialogflow plays a crucial role in implementing of the chatbot. Dialogflow is developed by Google for the sole purpose of creating Natural Language Processing applications. Dialogflow is a natural language understanding platform that makes it easy to design and integrate a conversational user interface into your mobile app, web application, device, bot, interactive voice response system, and so on. Using Dialogflow, you can provide new and engaging ways for users to interact with your product. Dialogflow can analyze multiple types of input from your customers, including text or audio inputs (like from a phone or voice recording). It can also respond to your customers in a couple of ways, either through text or with synthetic speech.

Creating a chatbot using Google Dialogflow contains the following steps:-

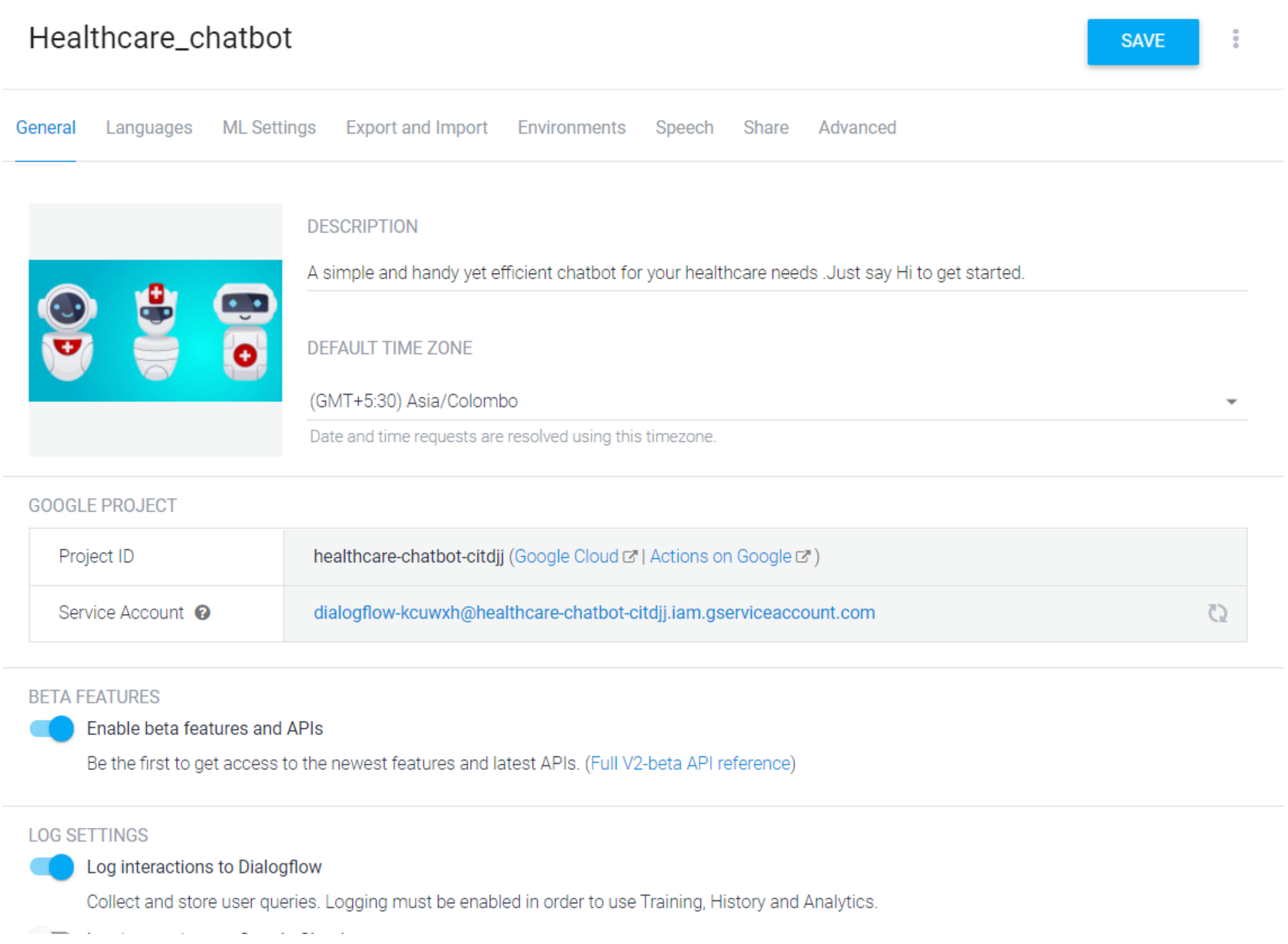
- Agent Creation
- Creating Entities
- Designing and developing intents
- Creating the knowledge base for chatbot
- Developing the fulfillment and designing the business logic
- Integration

All the steps involved in developing the chatbot are briefly discussed below.

Agent Creation

In this project we created a Dialogflow agent for the sole purpose of healthcare. A Dialogflow agent is a virtual agent that handles conversations with your end-users. It is a natural language understanding module that understands the nuances of human language. Dialogflow translates end-user text or audio during a conversation to structured data that your apps and services can understand. You design and build a Dialogflow agent to handle the types of conversations required for your system.
A Dialogflow agent is similar to a human call center agent. You train them both to handle expected conversation scenarios, and your training does not need to be overly explicit. The agent takes care of the flow of the conversation and also all the fallow-ups .

The below image shows how the agent creation page looks like and the parameters involved.



Healthcare_chatbot

SAVE

General | Languages | ML Settings | Export and Import | Environments | Speech | Share | Advanced

DESCRIPTION

A simple and handy yet efficient chatbot for your healthcare needs .Just say Hi to get started.

DEFAULT TIME ZONE

(GMT+5:30) Asia/Colombo

Date and time requests are resolved using this timezone.

GOOGLE PROJECT

Project ID	healthcare-chatbot-citdjj (Google Cloud | Actions on Google)
Service Account	dialogflow-kcuwxh@healthcare-chatbot-citdjj.iam.gserviceaccount.com

BETA FEATURES

Enable beta features and APIs

Be the first to get access to the newest features and latest APIs. (Full V2-beta API reference)

LOG SETTINGS

Log interactions to Dialogflow

Collect and store user queries. Logging must be enabled in order to use Training, History and Analytics.

Parameters involved:-
Agent name - Healthcare chatbot
Description - The small description of the chatbot that will be displayed on the deployment page to the user.
Project id – The Google cloud project id that is created to link the chatbot to the cloud.

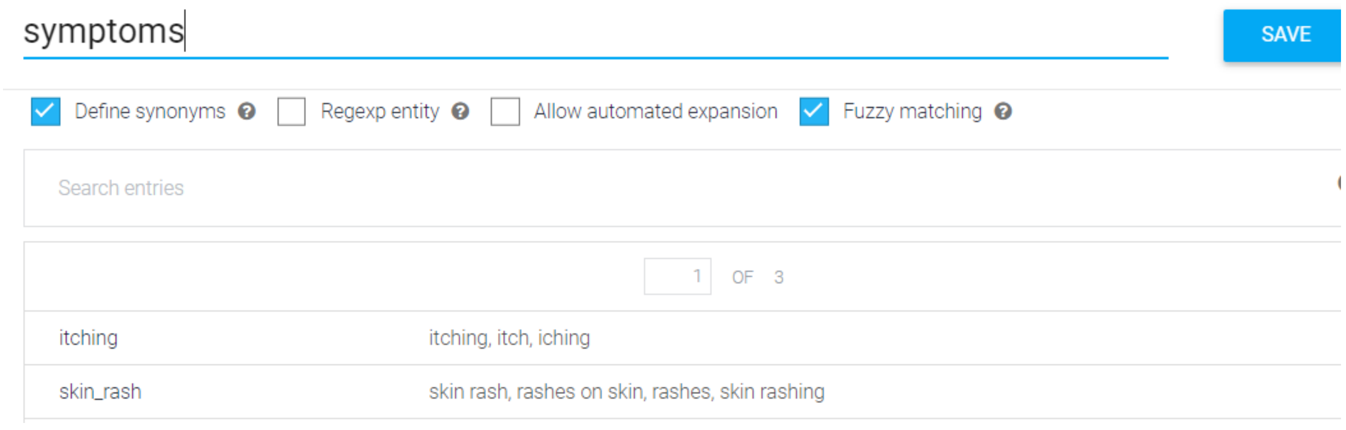
Creating Entities

Each intent parameter has a type, called the entity type, which dictates exactly how data from an end-user expression is extracted. There are primarily 2 types of entities that are system entities and custom entities. System entities are those entities which are predefined by the google cloud and commonly used. For example, there are system entities for matching dates, times, colors, email addresses, and so on. The system entities used in Healthcare chatbot are :-

- @sys.phone – entity for matching phone number

- @sys.any – entity for taking name input. By default there is a @sys.name entity in Dialogflow but it only recognizes western names and the primary audience of the chatbot will be Asian countries and so @sys.any entity is used.
- @sys.email – entity for matching email. If the email address entered does not meet the address expectations the error message is displayed and email address is again asked to enter.

The second type of entity is Custom entity. The custom entity is used for matching custom data related to the business logic. In this case the custom entity is created for symptoms.


symptoms

SAVE

[✓] Define synonyms · [ ] Regexp entity · [ ] Allow automated expansion · [✓] Fuzzy matching

Search entries

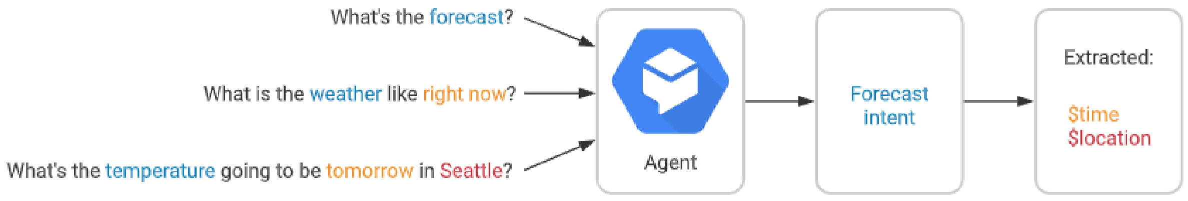
1 OF 3

itching	itching, itch, iching
skin_rash	skin rash, rashes on skin, rashes, skin rashing

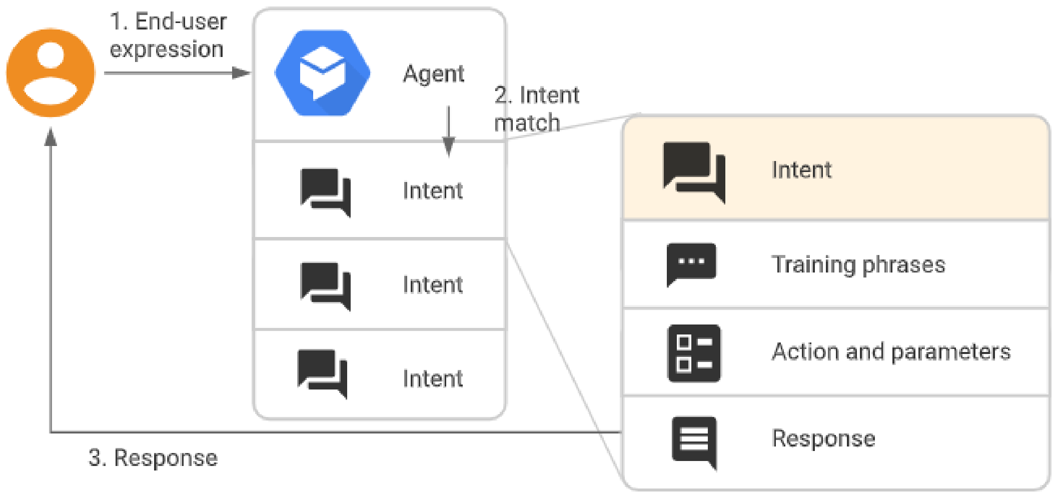
Synonyms are the words that a user may enter for the symptom. Left row contains the main symptoms that are to be understood by the agent and sent to the backend , right row contains the synonyms for the symptoms that a user might enter. Matching of these synonyms is of utmost importance and so **Fuzzy matching** is enabled. Using fuzzy matching, even if the user enters symptoms partially, the agent understands the intent of the user and acts accordingly.

Designing and developing Intents

An intent categorizes an end-user's intention for one conversation turn. For each agent, you define many intents, where your combined intents can handle a complete conversation. When an end-user writes or says something, referred to as an *end-user expression*, Dialogflow matches the end-user expression to the best intent in your agent. Matching an intent is also known as *intent classification*.
weather forecast. This extracted data is important for your system to perform a weather query for the end-user. In this case the primary intent of the user is disease detection and symptom classification. So each time a user enters the name of the symptom, Dialogflow agent matches the symptom name with the symptom entity and gets the exact symptom even if the user entered partial name of the symptom or there is a spelling mistake.


What's the forecast?
What is the weather like right now?
What's the temperature going to be tomorrow in Seattle?
Agent
Forecast intent
Extracted:
$time
$location

A basic intent contains the following:
<u>Training phrases</u>: These are example phrases for what end-users might say. When an end-user expression resembles one of these phrases, the agent matches the user expression with the intents due to machine learning enabled in the Dialogflow agent.
<u>Action</u>: You can define an action for each intent. When an intent is matched, Dialogflow provides the action to your system, and you can use the action to trigger certain actions defined in your system.
<u>Parameters</u>: When an intent is matched at runtime, Dialogflow provides the extracted values from the end-user expression as *parameters*. Each parameter has a type, called the entity type, which dictates exactly how the data is extracted. Unlike raw end-user input, parameters are structured data that can easily be used to perform some logic or generate responses.
<u>Responses</u>: You define text, speech, or visual responses to return to the end-user. These may provide the end-user with answers, ask the end-user for more information, or terminate the conversation.



In the following project, we developed simple intents for easy understanding by the user. Also all the symptom analyzing intents are linked to the analyzing initializer i.e. the intent which asks the user if he/she wants the quick diagnosis. If the user says/types yes, the agent asks the user what symptoms he/she is facing.

But the biggest question is how the agent knows that the "yes/no" entered by the user is for the question asked by the agent. This problem is solved setting the context of the intents. Dialogflow contexts are similar to natural language context. If a person says to you "they are orange", you need context in order to understand what they are referring to. Similarly, for Dialogflow to handle an end-user expression like that, it needs to be provided with context in order to correctly match an intent. Using contexts, you can control the flow of a conversation. You can configure contexts for an intent by setting input and output contexts, which are identified by string names. When an intent is matched, any configured *output contexts* for that intent become active. While any contexts are active, Dialogflow is more likely to match intents that are configured with *input contexts* that correspond to the currently active contexts.

Creating the knowledge base for the chatbot

The Knowledge base contains all the required dataset required for training and testing the machine learning model. It also contains the FAQ's for the chatbot that a user might ask.

Developing the fulfillment and designing the business logic

The fulfillment part of the chatbot is the part where we design the business logic i.e. the backend coding for the chatbot. There are multiple ways to deploy the backed for the chatbot.

1. **Webhook**:- Webhook is the service where a custom API can be called to take care of the backend of the chatbot.
   The following requirements must be met by the webhook service:-
   - It must handle HTTPS requests. HTTP is not supported. If you host your webhook service on Google Cloud Platform using a Compute or Serverless Computing solution, see the product documentation for serving with HTTPS.
   - Its URL for requests must be publicly accessible.
   - It must handle POST requests with a JSON WebhookRequest body.
   - It must respond to WebhookRequest requests with a JSON WebhookResponse body.

2. **Inline Editor**:- Inline editor is the IDE provided by the google cloud itself where you can write the business logic. The inline editor only supports Node.js and uses the Dialogflow fulfillment library. When we initially enable the inline editor, the fulfillment code is pre-populated with default handlers for default intents that are included for all agents. The code also has commented instructions for adding handlers for developer-defined intents. The inline editor is intended for simple fulfillment testing and prototyping. Once we are ready to build a production

application, we can create a webhook service.

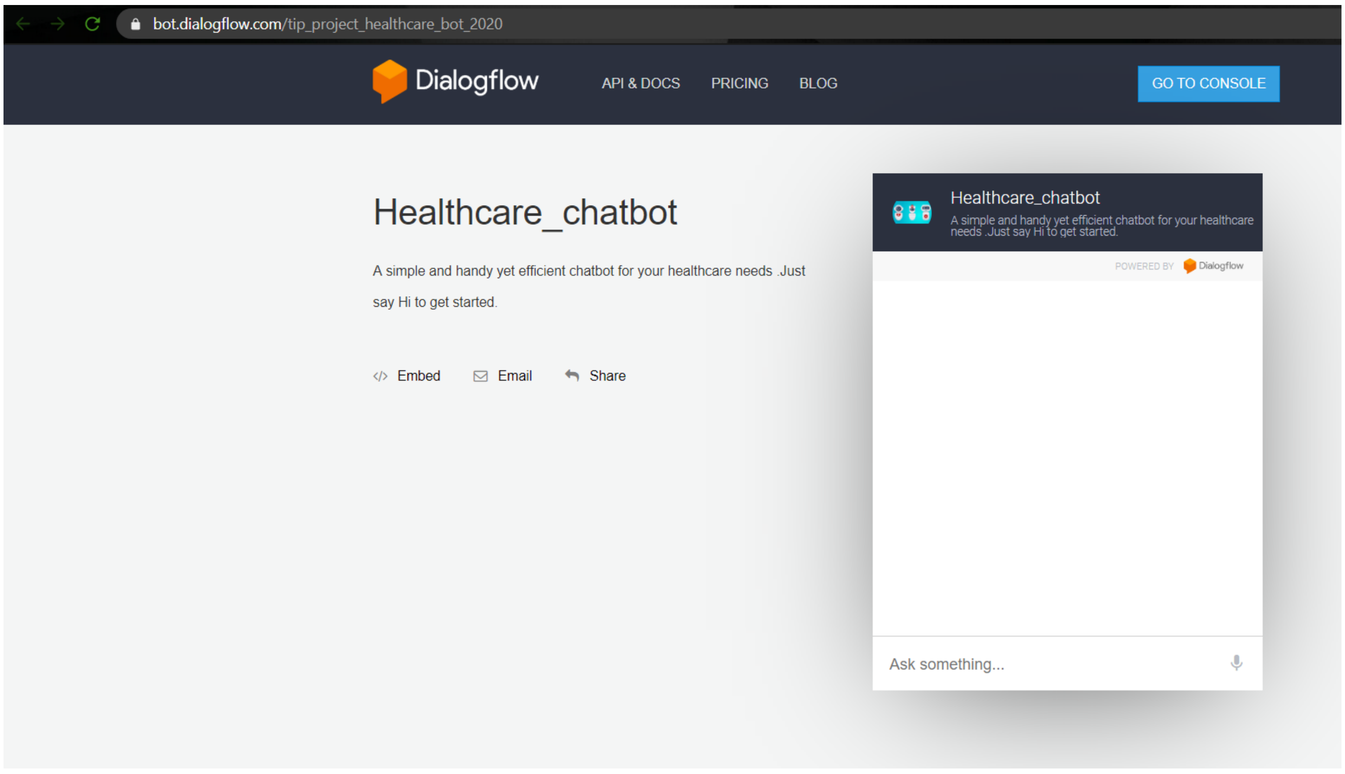
For this project, we used Heroku cloud for the deployment of the business logic. Heroku cloud is a free cloud hosting service where we can deploy the cloud functions and using this the time complexity can be reduced and also the cloud function charges can be reduced .

Integration

Integration is the last and final part of the chatbot deployment. With an integration, end-user interactions are handled for the chatbot.

The chatbot is integrated with :-

1. The Dialogflow cloud where the Dialogflow takes care of the agent. It provides a webpage where a user can access the deployed chatbot very efficiently. Using the following link the chatbot can be accessed https://bot.dialogflow.com/tip_project_healthcare_bot_2020



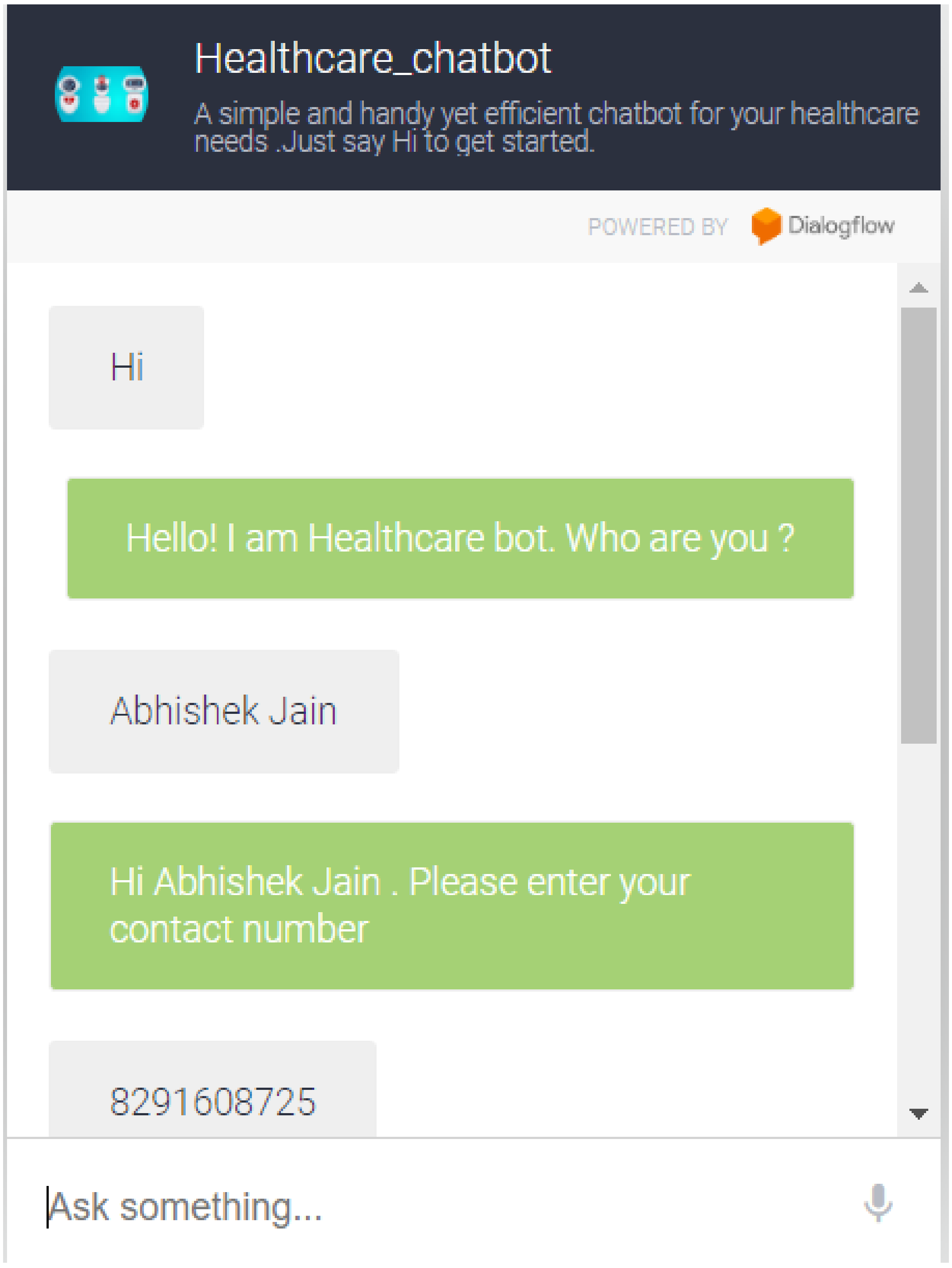
Healthcare_chatbot

A simple and handy yet efficient chatbot for your healthcare needs .Just say Hi to get started.

2. Telegram – Telegram has a chatbot integration with the name “ProjectHealthcarebot”. With the help of “botfather” which is a service offered by telegram, the chatbot is easily deployed on telegram as well. Just by searching “ProjectHeatthcarebot” one can easily access the chatbot.

3. Other platforms:- We are also planning to deploy the chatbot on various other platforms such as “whatsapp”, “facebook messenger”, “slack” etc.

OUTPUT



Healthcare_chatbot

A simple and handy yet efficient chatbot for your healthcare needs .Just say Hi to get started.

POWERED BY Dialogflow

Hi

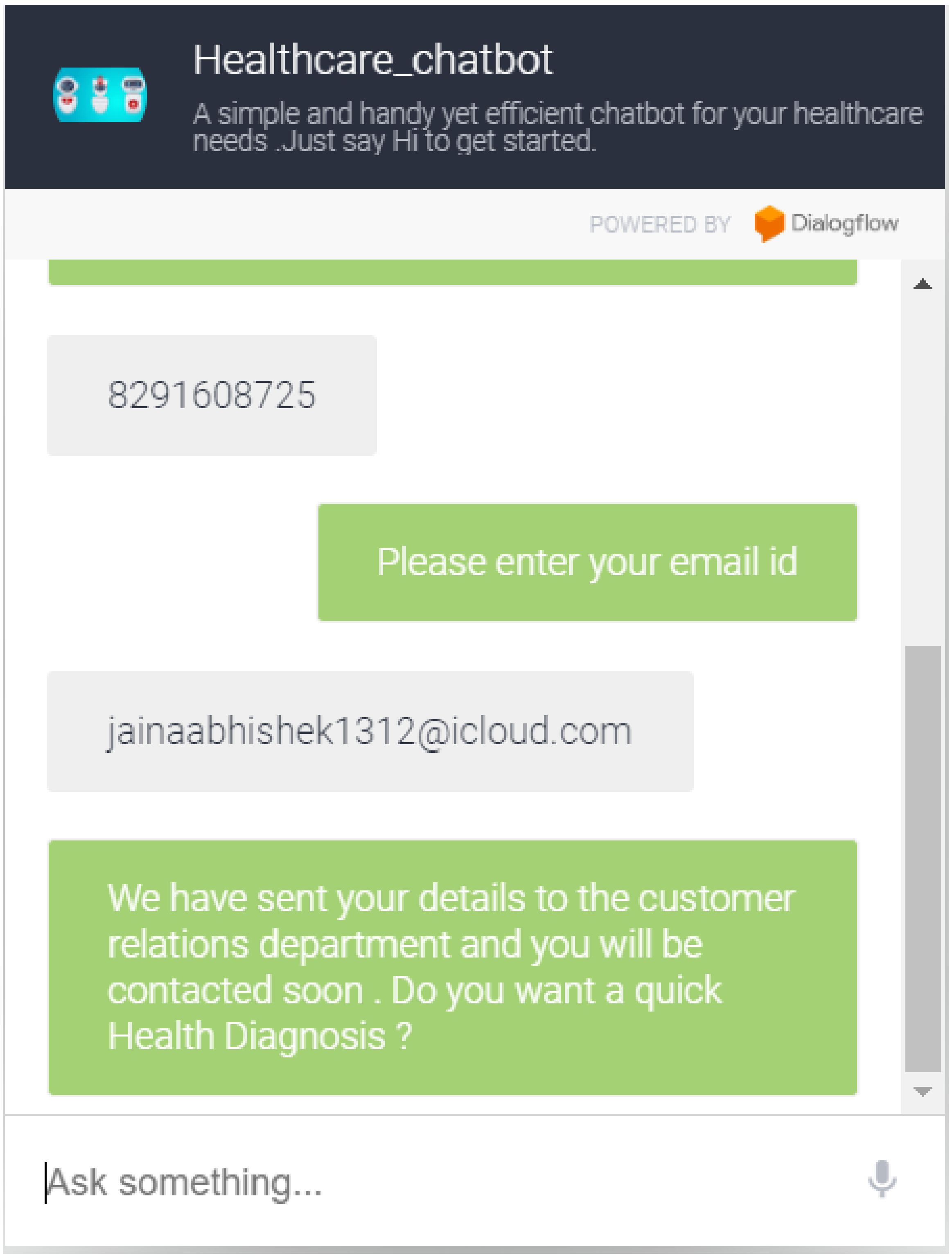
Hello! I am Healthcare bot. Who are you ?

Abhishek Jain

Hi Abhishek Jain . Please enter your contact number

8291608725

Ask something...



Healthcare_chatbot

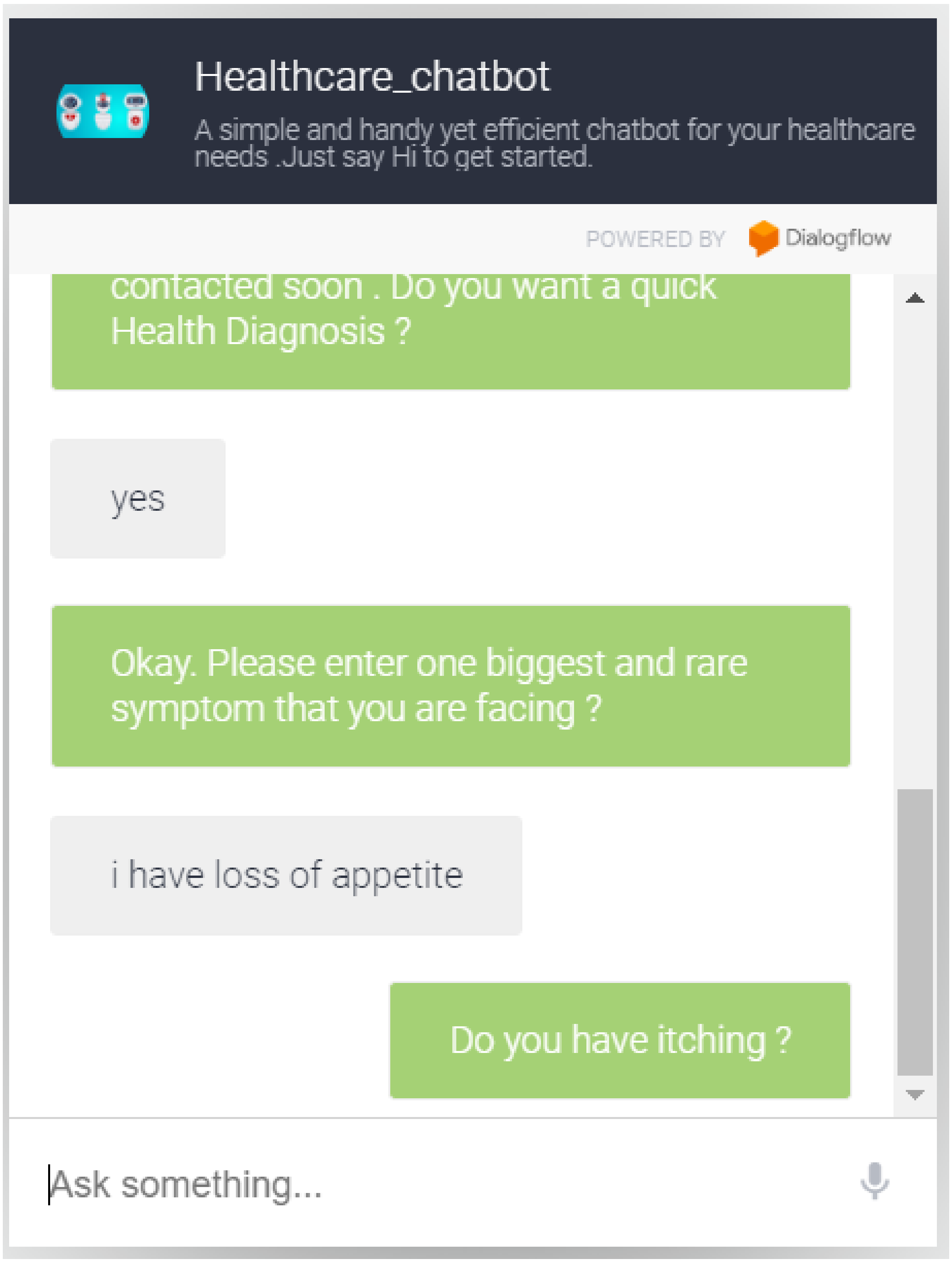
A simple and handy yet efficient chatbot for your healthcare needs .Just say Hi to get started.

8291608725

Please enter your email id

jainaabhishek1312@icloud.com

We have sent your details to the customer relations department and you will be contacted soon . Do you want a quick Health Diagnosis ?



Healthcare_chatbot

A simple and handy yet efficient chatbot for your healthcare needs .Just say Hi to get started.

POWERED BY Dialogflow

contacted soon . Do you want a quick Health Diagnosis ?

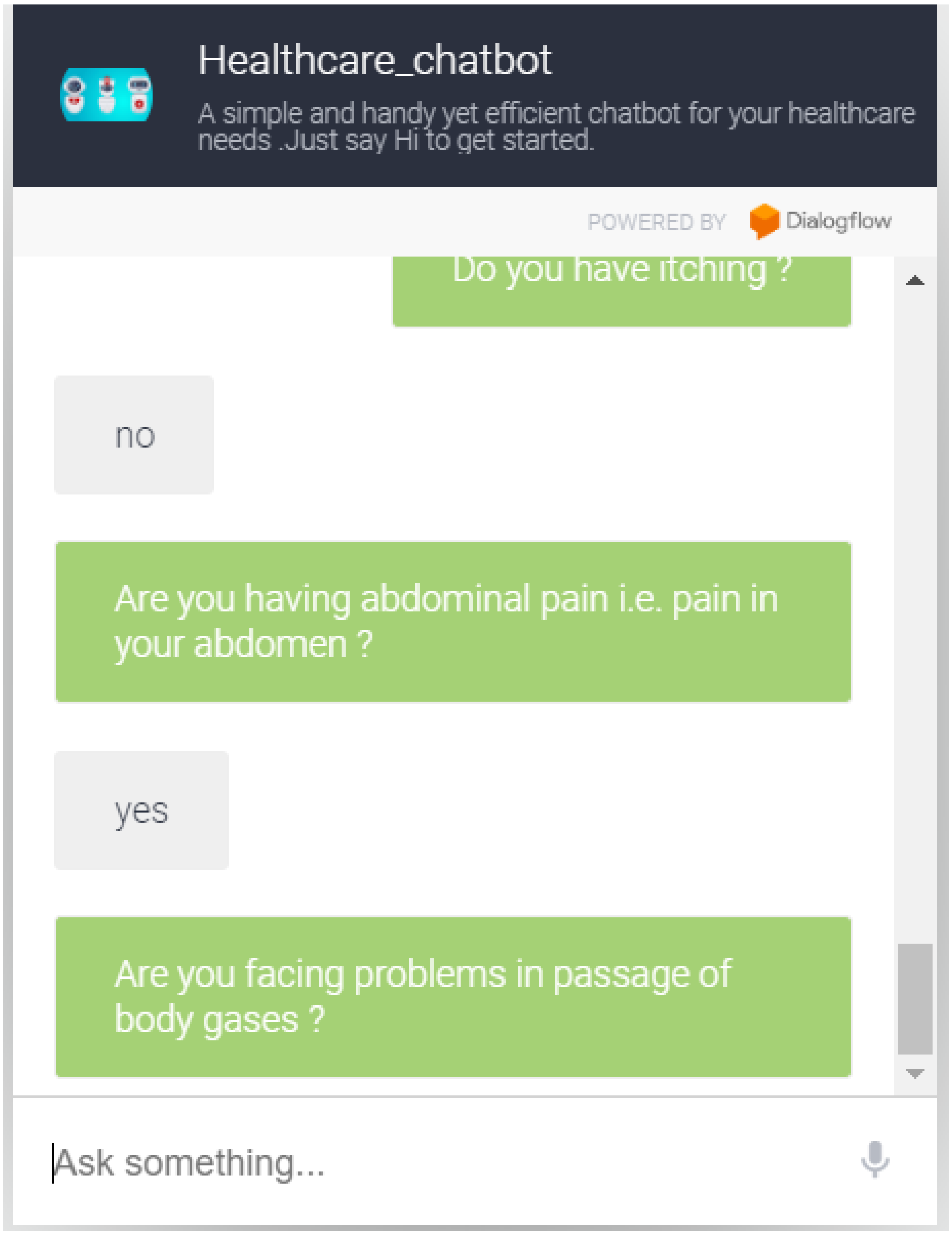
yes

Okay. Please enter one biggest and rare symptom that you are facing ?

i have loss of appetite

Do you have itching ?

Ask something...



Healthcare_chatbot

A simple and handy yet efficient chatbot for your healthcare needs .Just say Hi to get started.

POWERED BY Dialogflow

Do you have itching ?

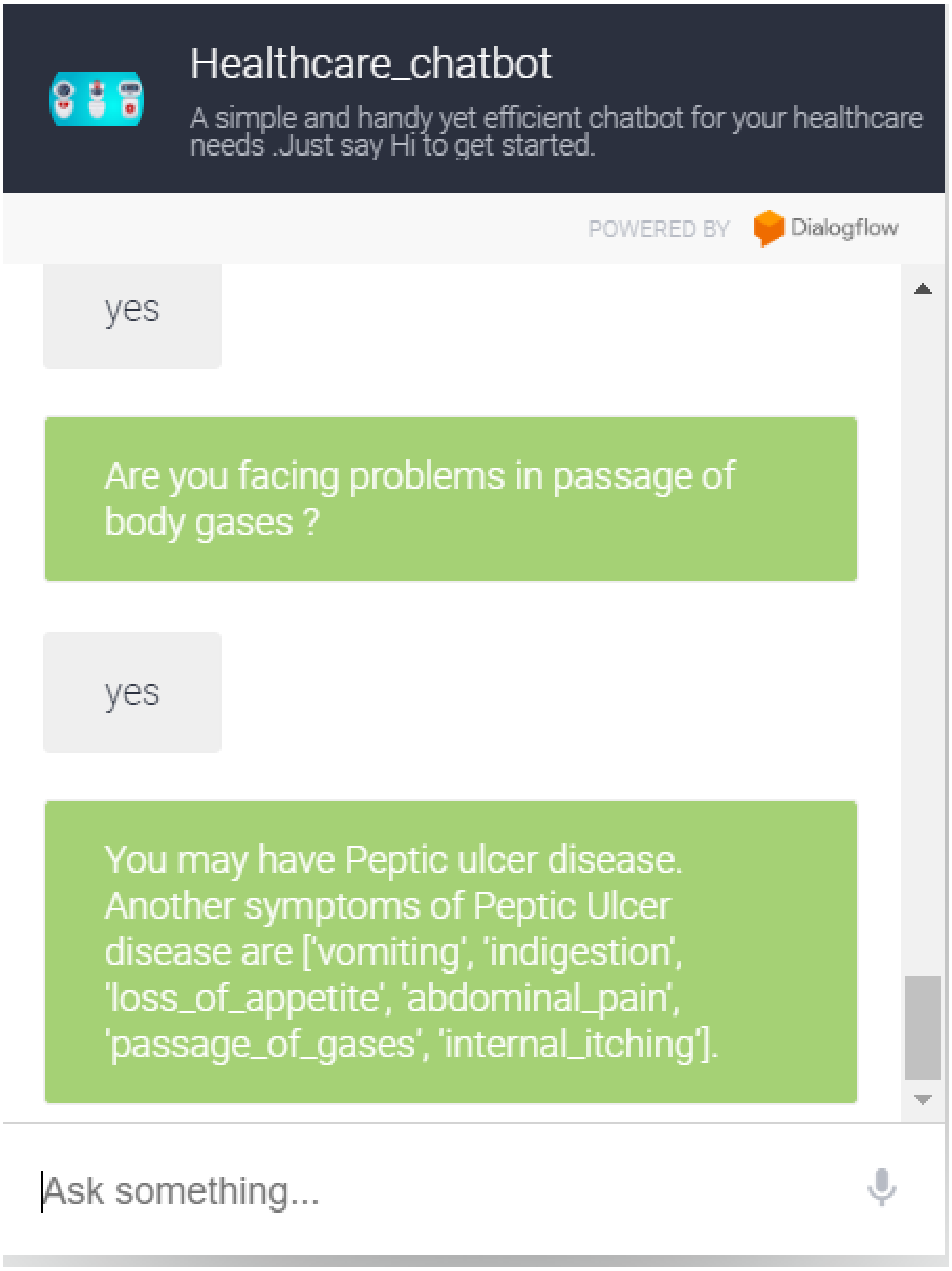
no

Are you having abdominal pain i.e. pain in your abdomen ?

yes

Are you facing problems in passage of body gases ?

Ask something...



Healthcare_chatbot

A simple and handy yet efficient chatbot for your healthcare needs .Just say Hi to get started.

yes

Are you facing problems in passage of body gases ?

yes

You may have Peptic ulcer disease. Another symptoms of Peptic Ulcer disease are ['vomiting', 'indigestion', 'loss_of_appetite', 'abdominal_pain', 'passage_of_gases', 'internal_itching'].

CONCLUSION

We tried to make a Natural language processing bot which works on Google cloud platform and written in python. We tried to make it as interactive as possible but due to the Google cloud policies we didn’t have sufficient resources to make it more interactive and feature loaded as the next generation chatbots. The chatbot can be directly accessible over the internet using the link provided and is very easy to use and understand. The voice input/output is also supported using the google natural language understanding engine. I would like to conclude by saying that this project helped me in understanding various new technologies such as Google cloud platform, Natural language processing, etc which will help me further to contribute something fruitful to the rising industry of cloud platforms.

REFERENCES

- www.cloud.google.com
- www.concole.dialogflow.com
- www.cloud.google.com/dialogflow/docs
- www.towardsdatascience.com
- www.wikipedia.com