

# DASH

Diagnosis and Symptoms Helper  
An intelligent triage and pre-diagnosis system

## **Team members**

Tadhg Kennedy

Meng Chenxi

Wang Hongtao

Yang Yizhou

November 3, 2021

# Contents

<b>1</b>	<b>Executive Summary</b>	<b>4</b>
<b>2</b>	<b>Business Case</b>	<b>4</b>
2.1	Problem Statement . . . . .	4
2.2	Proposed Solution . . . . .	4
2.3	Market Research . . . . .	5
2.3.1	Doctor Interview . . . . .	5
2.3.2	Survey Results . . . . .	5
2.4	Project Team . . . . .	6
<b>3</b>	<b>Product Design</b>	<b>6</b>
3.1	Main Features . . . . .	6
3.2	Process Flow . . . . .	6
3.3	Systems Highlights . . . . .	7
3.3.1	Short-term memory based chatbot . . . . .	7
3.3.2	Natural Response . . . . .	7
3.3.3	Extensible software framework design . . . . .	7
<b>4</b>	<b>System Modeling</b>	<b>8</b>
4.1	Data Preparation . . . . .	8
4.2	Model Choosing . . . . .	8
4.3	Model Testing . . . . .	9
<b>5</b>	<b>System Development &amp; Implementation</b>	<b>9</b>
5.1	User Interface . . . . .	9
5.2	Chatbot . . . . .	9
5.2.1	Dialogflow . . . . .	9
5.2.2	Questioning Strategy . . . . .	10
5.3	Backend Process . . . . .	10
5.3.1	User Information Management Process . . . . .	10
5.3.2	Ngrok with chatbot integration . . . . .	11
5.4	Conclusion . . . . .	12
5.4.1	Future Development . . . . .	13
<b>A</b>	<b>APPENDIX A - Project Proposal</b>	<b>15</b>
<b>B</b>	<b>APPENDIX B</b>	<b>18</b>
<b>C</b>	<b>APPENDIX C - Installation &amp; User Guide</b>	<b>18</b>
<b>D</b>	<b>APPENDIX D - Doctor Interview</b>	<b>22</b>
<b>E</b>	<b>APPENDIX E - Individual Report</b>	<b>22</b>
E.1	Personal Report: Tadhg Kennedy . . . . .	23
E.1.1	Personal Contribution . . . . .	23
E.1.2	Lessons Learned . . . . .	23
E.1.3	Future . . . . .	23
E.2	Personal Report: Wang Hongtao . . . . .	24
E.2.1	Personal Contribution . . . . .	24
E.2.2	Lessons Learned . . . . .	24
E.2.3	Future . . . . .	24
E.3	Personal Report: Meng Chenxi . . . . .	25
E.3.1	Personal Contribution . . . . .	25
E.3.2	Lessons Learned . . . . .	25
E.3.3	Future . . . . .	25
E.4	Personal Report: Yang Yizhou . . . . .	26

E.4.1	Personal Contribution . . . . .	26
E.4.2	Lessons Learned . . . . .	26
E.4.3	Future . . . . .	26

# 1 Executive Summary

The COVID-19 pandemic has changed the way we approach doctor's visits and medical institution as a whole. With a highly transmissible disease out in the population having sick people congregate in a location is the exact opposite of what doctors and immunologists advise. While the COVID-19 pandemic is waning the impacts it has had on our approach to medicine can be seen across the industry, tele-medicine services are now common place and efforts to reduce patient congregation are still in place to reduce the chances of a potential spike.

Our diagnosis and symptoms helper (DASH) seeks to address this need and provide additional benefits to the institutions it's deployed in. By leveraging the power of machine learning and chatbots we hope to be able to reduce the need for patients to congregate in waiting rooms and reduce some of the workload from the doctors in their already busy workdays. Our system will collect the patient symptoms via a chatbot, these symptoms will then be used to intelligently generate questions to help narrow down the possible diagnoses, once the system has sufficient information the symptoms will be used to provide a potential diagnosis in a report to the doctor. This report will allow the doctor to decide if the patient requires an in person consultation, tele-medicine session, provide prescription or potentially skip the visit to the GP and proceed straight to a specialist.

This project will provide a minimum viable product (MVP) for the system, focusing on the core functionality in order to deploy, test in a real world scenario, and gather feedback to help improve the product. The product has the potential to grow to encompass many useful functionalities such as prescription recommendation, automated booking systems, and improved diagnosis based on the patients medical history.

## 2 Business Case

### 2.1 Problem Statement

With COVID still a major concern, people congregating while waiting to see the doctor increases the risk of transmission to other potentially vulnerable patients. Some patients may even have illnesses which can't be treated with antibiotics or prescription medications, such as cold or flu, in these cases it would be better that the person does not come to the clinic and risk exposure, and instead remains at home to rest and recuperate.

Additionally health care workers are stretched thin and over-worked due to the demands of the pandemic, this can lead to overwhelming workloads, mounting backlogs and eventually burnout. Further reducing the manpower available at a time where it is so desperately needed.

### 2.2 Proposed Solution

Our proposed solution to tackle these issues is to use a AI chatbot to gather patients symptoms before they go to the clinic in person, intelligently generate questions to further narrow the potential diagnosis before finally generating a report for a doctor to review and make recommendations for how the patient should proceed. The system will focus on acute illnesses which can be pre-diagnosed by unquantifiable symptoms (sore throat, itchiness, rash).

The doctors can then recommend: tele-medicine services for those with illnesses which don't require further in person tests and prescribe medication as necessary, in person consultation for illnesses which require test like blood pressure or samples, or potentially direct referral to specialists for time sensitive or severe illnesses.

Our solution has the following objectives

1. **Reduce unnecessary visits to clinics** Help patients and doctors avoid visiting clinics and potentially exposing themselves to infectious disease.
2. **Assist doctors** Reduce some of the workload from doctors by collecting symptoms in advance and providing possible diagnosis.
3. **Reduce wait times** By reducing the number of people visiting clinics in person, patients who need it can be seen quicker and potentially skip clinic visits to be referred to a specialist saving precious time in cases of critical illness.

## 2.3 Market Research

AI healthcare is a rapidly growing market, reaching \$6.7 Billion in 2021 [Kwo21] and is forecast to reach a market size of USD 61.59 Billion by 2027 [RD21]. This ever expanding industry along with Singapore's commitment to the Smart Nation projects in the medical sector show that AI is a major focus for Singapore's future.

Based on our research we have found there are several other products which attempt to provide medical diagnosis via chatbots and apps. However, these products are either focused on providing the diagnosis and suggested treatments to the patient or to gather information about the patient before sending them to the companies own private tele-medicine services [Fut17]. We intend to focus on provided value to existing brick and mortar locations and the public sector with our product, we believe this approach will provide the best value for the community and work best in Singapore due to the nationalised healthcare system.

We have also conducted our own market research starting by interviewing a doctor to gather information about their day to day work in a polyclinic here in Singapore and to get their opinions of the usefulness and viability of our idea. Next we surveyed patients globally and doctors both in Singapore and in China, using the information we gained from the interview to decide what questions to ask. The surveys were split into two parts: for patients the first was to gather their general experiences with doctors and approach to illness, second was to get an understand of their exposure to AI and tech in medicine and hear their concerns; for doctors the first was to understand their general day to day work and experience with patients, second was to their experience with AI in medicine, what they want from a product like ours and what concerns they have.

### 2.3.1 Doctor Interview

We were lucky enough to be able to interview Dr. Li Kim Selby, a two year family medicine resident at the Geylang Polyclinic. She was kind enough to take the time to speak with us and answer our questions, from this we were able to get a general estimate of how many patients with acute cases would be seen in a day along with an understanding of how long doctors had to deal with these patients and the percentage of patients who didn't need prescriptions for their illness. She also explained that AI isn't something she uses in her day-to-day but sh does believe it could help. She did however raise concerns over the model creating a confirmation bias with doctors, we have taken this feedback under advisement and based on the results from our MVP test will investigate potential solutions to combat this. The full set of interview questions is available in the appendix D, we'd like to thank Dr. Selby for her time in helping guide the design of DASH.

### 2.3.2 Survey Results

**Patients Survey** Based on our survey with 73 respondents, we were able to get a general feeling for the public's interactions with doctors and opinions about AI in medicine. Our sample was 66% over 45 years of age, with approximately the same split in the age date, most visited the doctor fewer than 5 times a year. 70% of the respondents had not used a tele-medicine service, 53% of respondents would not engage with an AI over a doctor, with 30% being neutral and the remaining 17% being willing. General concerns were primarily accuracy with 75% citing this as their major concern. Based on these responses we feel that we have made the correct decision is designing the product to provide a report to the doctor, as this will address any of the fears raised by the public over an AI making the final decision.

**Doctor Survey** We received 16 response from doctors in Singapore all within the GP or family medicine discipline, across many years of experience ranging from 3 to 34 years. We gathered a further 22 responses from doctors and medical students in China.

65% of responding doctors had not used AI as part of their work, while 35% had, this shows that AI is still an emerging field in medicine. We can also see from the results that the doctors with the most experience were the least likely to trust AI, while younger doctors were more neutral, this may be due to an unfamiliarity with what AI is for older doctors or past experience with less advanced AI systems. 68% of responding doctors indicated that a chatbot to collect patient symptoms would be helpful in their day to day work. With the 57% of respondents taking 3-5 minutes to collect symptoms per visit, and 20-40 visits per day, our system could save between 1 to 2.5 hours a day for the doctor.

The legal implications of engaging an AI was also raised by some doctors in the comments, as previously stated we’ve chosen to design the system as an assistant for doctors to make the final diagnosis and in all cases the patient will be seen by a doctor which avoids any potential occurrences of misdiagnosis of critical illnesses. When the system is deployed we will work to ensure the system strictly follows the guidelines set out by the MOH regarding AI in medicine. [oH21]

## 2.4 Project Team

Full Name	Roles
Tadhg Kennedy	Administrative lead Market Research Intelligent question generation Marketing Video Project report writing
Meng Chenxi	Model development and testing Project report writing
Wang Hongtao	UI design and development Backend Function Project report writing
Yang Yizhou	Technical Lead Chatbot development Backend framework development Project report writing

## 3 Product Design

### 3.1 Main Features

Just like the results of our surveys, we can find that main pain points are that patients often have to spend a lot of time before they get a treatment and when AI is involved in treatment, it will trigger the distrust of doctors. Therefore, we mainly propose the following features.

- **A chatbot that collects patient information.**

Because the cornerstone of diagnosis in the modern healthcare system is trust. From this perspective, we don’t let the user notice too much the presence of AI in the diagnosis, so we use a chatbot for patient information capture.

- **An intelligent diagnostic system.**

Doctors often use symptoms to speculate about illnesses, and the reasoning is often based on medical knowledge and previous cases. So we can reduce it to a process of inductive reasoning, also called knowledge representation and search. In this way, we can give enough possible conditions for the doctor to make a diagnosis, and even some conditions that the doctor has not considered.

- **An app for doctors.**

We also designed an application for doctors to search for patients information and use AI to advise on the disease and even provide referable treatments.

### 3.2 Process Flow

Figure 1 shows the workflow of this system. It can be found that the system provides two external interfaces, one is a chat window in the form of Chatbot, which is mainly provided to patients. One is an interface in the form of a desktop application. Through this interface, doctors can query, update, modify, and delete entries in the database.

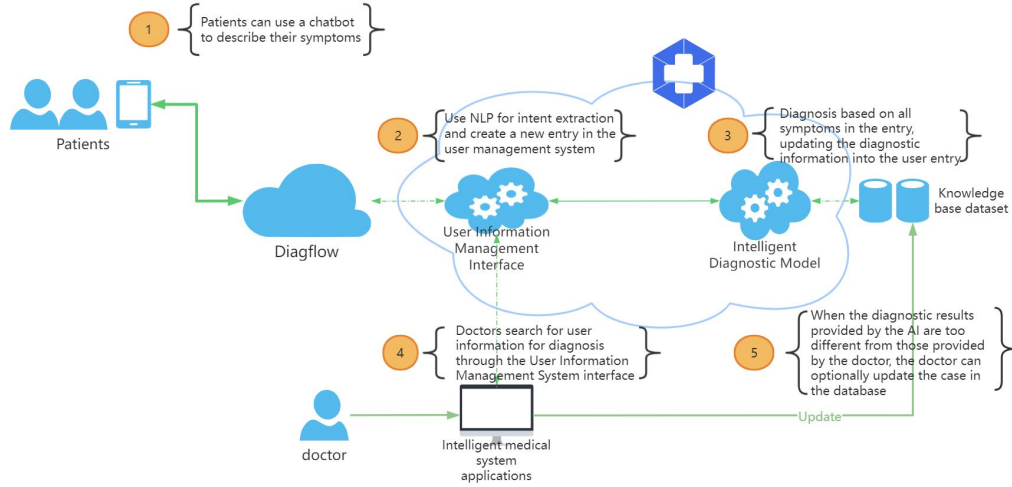


Figure 1: Process Flow

### 3.3 Systems Highlights

#### 3.3.1 Short-term memory based chatbot

Our intelligent medical system is a short-term memory base system which can remember the symptoms described by the patient and narrows them down to a specific symptom based on the results of the next questioning, for example, if the patient says, "I have a fever," we will write down the symptom of fever, but it is uncertain whether the fever is high or low, and we will narrow down the symptom in the next round of Ask, "Is it a high fever or a low fever", to narrow down the symptoms.

#### 3.3.2 Natural Response

We use a template-based response generation module to respond to patient questions. The input to this module is a specific symptom and the output is a natural flowing sentence as a response to the patient's question.

#### 3.3.3 Extensible software framework design

Figure 2 shows our software architecture design, which can be seen in three main modules, user information management module, core module, and function module. This framework is designed to keep the modules separate from each other. Modules and databases are independent of each other, with high cohesion and low coupling and easy extension, making it possible for each member of the team to do some of the work independently.

- **Core module**

The core module has two main processes. One process acts as the back-end process for Diagflow, storing, filtering and processing the intent returned by Diagflow. The other process is mainly used to provide a visualization application for the doctor to view the patient's specific situation.

- **User information management module**

User Information Management Module contains a set of API for operating the database and a processing process. The API is mainly used by other modules to safely operate the database, and the process monitors the data submitted by other modules continuously and calls functions in the function module to process the data that needs to be processed.

- **Functional module**

Functional module is designed to be extensible, now we have implemented the diagnosis function and subsequently if you want to implement more functions in this framework can also be easily added.

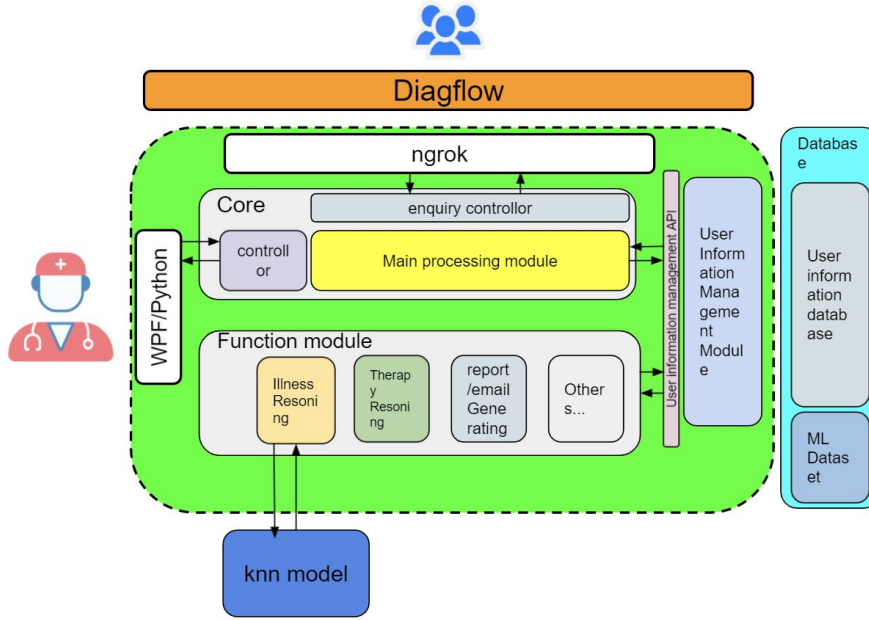


Figure 2: Framework

- **Exterior Design**

Exterior Design is similar to a plug-and-play design, which means we can design and train the models independently, update the database, and then replace them with the original model and database and then they will work fine.

## 4 System Modeling

### 4.1 Data Preparation

We collect the data on Kaggle, and each piece of data is a patient's symptoms and its disease. We do the preprocessing to the dataset and divide it into training set and test set, which will be convenient to our later work. There are 132 symptoms and 41 diseases in this training set, and 4920 patients to train the model. The testing set contains only 42 pieces of data but covers all disease so we won't miss one single situation. The details of our datasets like this: the symptoms which patients have are marked as 1 and other symptoms are marked as 0. The last column records the diseases of each patients. The symptoms are very common and detailed, such as headache, cough and high fever. The diseases range from simple cold to AIDs.

### 4.2 Model Choosing

We look up some papers and found that most similar projects use KNN(k-nearest neighbors)[ESG16][XB19] and DT(decision tree)[CCH19] to build the model. Therefore we decide to try both of them and then find the better one.

We first use KNN to build the model and the function 'knn.score' to check its performance. We set the number of neighbors which is 'k' in the algorithm from 20 to 100, and record the computing time. The most suitable k is about 50, which has a high score of 94.9% on the training set and 92.9% on the testing set. At the same time, the computing time is acceptable which is important for several times of training and we believe it's a useful point in real world case.

Then we use DT to build the model and the function 'dt.score' to check its performance. We change the parameter 'max.depth' to find the best depth of DT and find 38 is the most suitable depth, which has a score of 91.8% on the training set and 90.6% on the testing set and the computing time of DT model is much longer than KNN model. Moreover, based on the actual problem of this case, it's



difficult for DT to get the next symptom during the talk with patient. At last, we decide to choose KNN model and set  $k$  as 50. We train KNN model on the training set and get the final model quickly and then save it for future use.

Now that we have the trained model, the next step is to build other functions to complete the process. Here is how the system works: we first get the symptom that user provides and use model to predict potential illnesses and their probability. Then we pick symptom with highest fluency in those potential illnesses and set it as the next question to be asked. All symptoms that user gives yes answer will be marked as 1. Once the sequence of potential illness come to one single illness or all symptoms have been asked, the chat will come to an end and the system will save the final result.

### 4.3 Model Testing

We start to test it using specific illness. Here I take common cold as an example.

We first collect all 17 symptoms of common cold in the training data and create a sequence to simulate user's input. First we set 'cough' as 1 while other symptoms as 0, which means the first user input is 'cough'. Second we put this sequence into our model and get a result of potential illnesses which are 'Bronchial Asthma', 'Common Cold' and so on. Moreover, we can get the symptoms to be asked next which is 'chest pain', but it don't match 'common cold' so we skip it and find another symptom which is 'high fever'. Same as the last step, we set 'high fever' as 1 and put the sequence into model to make prediction. Mostly, the number of potential illnesses will reduce step by step and finally reach the predicted result. After we integrate the whole system, we use diagflow to do the same test and at the same time check the outputs in the command window and find the same result.

During the course of the testing we had to tackle many problems. Initially we found that the distance function being used didn't yield the correct results. Through experimentation we were able to find that the "russellrao" setting yielded much better accuracy. We also found that initially the way the questions were being generated wasn't a very good experience for the patient as in some cases the chatbot would end up asking questions for too long. Investigating in the backend we found that even when the diagnosis was certain if there were many symptoms associated with the illness the chatbot would ask all of them even if it didn't help to increase the accuracy. We made updates to the logic such that when the diagnosis is narrowed to only one possible outcome the chatbot stops asking further questions, this made the experience much better for the patient and increased efficiency.

## 5 System Development & Implementation

### 5.1 User Interface

Initially, we decide to develop a UI system for doctor to see information of patients including patient user information, the symptoms they provided and the illness that our chatbot had predicted.

The front end was developed using pyqt. This system was divided into three parts: search, view and exit.

Once the doctor enter into this app, it will display search page and required to input patient id or name in order to get the information from the database and then click the "submit" button, it will then show the view page to display the symptoms and illness that our chatbot predicted.

Figure 3 show the search page, which is based on the id or name to search the information of patients.

Figure 4 show the view page, which can display the symptoms and illness that our system collected and predicted.

### 5.2 Chatbot

#### 5.2.1 Dialogflow

Dialogflow is a natural language understanding platform used to design and integrate a conversational user interface into mobile apps, web applications, devices, bots, interactive voice response systems and related uses.

**Entities Design.** First we divided the different symptoms into parts and symptoms, for example, "eye\_pain" is a symptoms in the dataset, so we divided it into "eye" part of our body and "pain" the

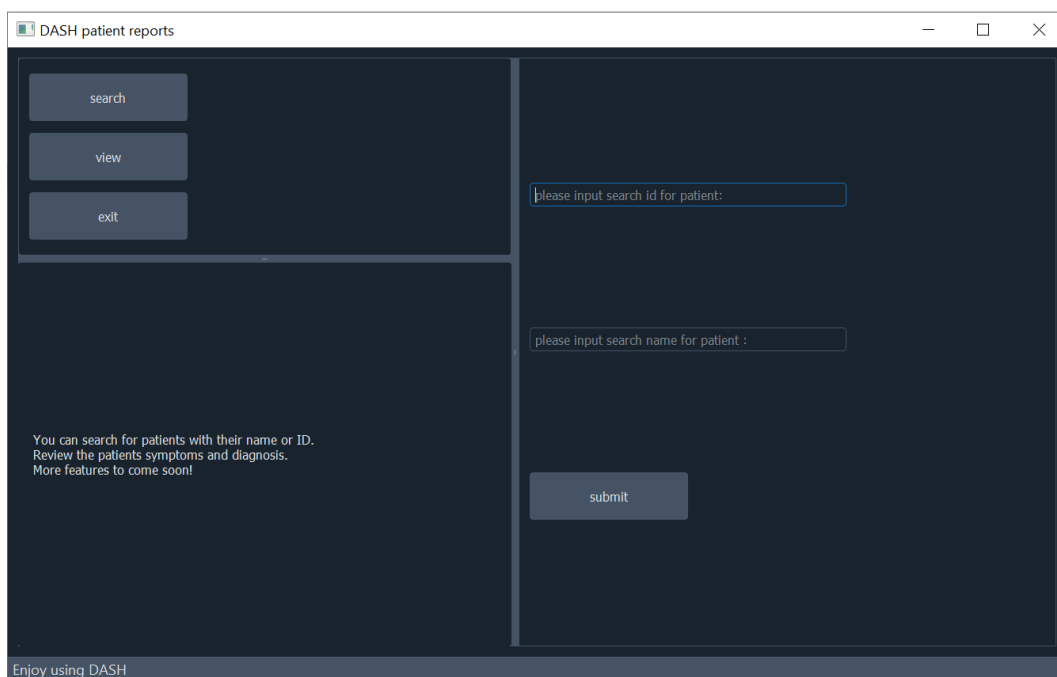


Figure 3: UI Search page

description of this symptom. As analyzed, we created two new entities and divided all the symptoms in the dataset into these two entities, and added some synonyms to make the identification more accurate, like "fever" and "high temperature".

**Intents Design.** Dialogflow can help us extract intents form a sentence. In this task, the main intention is to identify two intentions: 1. the illness described by the patient, which we named "illness", and 2. the attitude of the patient in answering the question, whether it is positive or negative. For example when the patient says "I have a fever", the specific symptom "fever", will be returned by Dialogflow to the system, and when the patient answers a question generated by the system in the chat window, we can determine if they answered yes or no.

### 5.2.2 Questioning Strategy

We want the chatbot to automatically respond more naturally, even like a real person communicating with a patient, so the logic of asking is essential. Based on our daily query experience, we divided the query logic into three steps.

Figure 5, you can find that we summarized our daily experience, firstly, describing the condition and doctor questioning are relatively well understood, and it is worth mentioning that we designed step 2 to narrow down the symptoms based on short-term memory. For example, if in the first step of describing the illness, the patient says his eyes feel uncomfortable, but does not express his specific symptoms. And then we will search all the symptoms related to the eyes in the database and ask the patient with these symptoms, then we achieve the purpose of obtaining specific symptoms. In the third step of doctor questioning, we used the KNN model to get the possible illnesses under the asked symptoms and searched the symptom common to these illnesses as the next question asked by the chatbot.

## 5.3 Backend Process

### 5.3.1 User Information Management Process

Before describing this process, first we need to explain the data structure in the database.

Figure 6 shows the data structure of the user information in the database, including information on the 142 symptoms, as well as the predicted conditions and corresponding treatments.

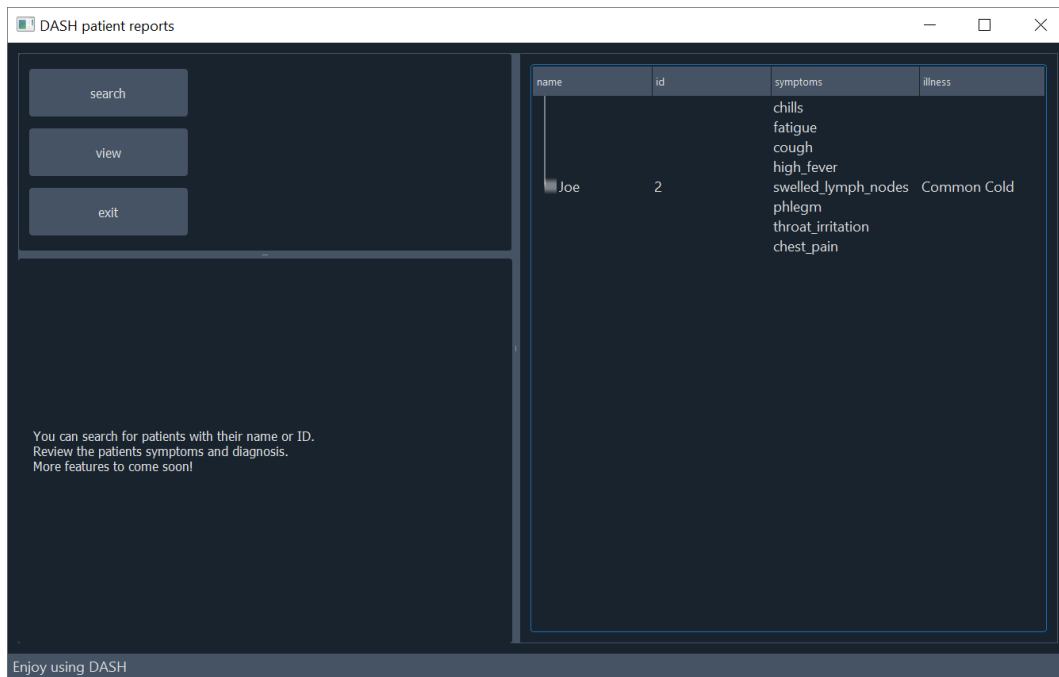


Figure 4: UI View page

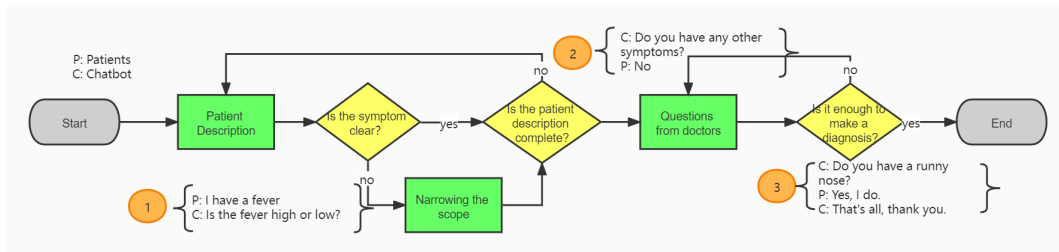


Figure 5: Query Logic

The user information management system (UM) provides a series of interfaces for other such as UI, ngrok to make changes to the user information database.

In Figure 5, you can notice that unlike the usual API design, behind the interface for adding, deleting, and checking, we designed a process to handle the submitted entries. The reason for this design is that the entry submission is not simply a modification to the database, but requires a call to 'get\_nextquestion' to get the next question asked and also 'get\_illness' to get the most likely current illness and record it to the database, both functions require the involvement of the model.

Figure 8 shows the processing of data by the user information management process (UM for short). First UM makes a judgment on the owner field, and only processes this entry if it is judged to belong to UM. After that specific functions will be called to populate the entry, for example the illness field, UM will iterate through all the symptoms with the value of 1 and then input them into the model, after getting the illness predicted by the model, we input the illness into the illness field.

### 5.3.2 Ngrok with chatbot integration

The main questioning strategy has been covered in section 5.2.2, and here we focus on how chatbot is implemented in software.

For a chatbot, the input is a sentence from the user and the output is a reply. In our framework above, Diagflow has helped us extract the intent from the input sentence, so the remaining step is to generate a reply based on these intents.

Figure 9 illustrates the data structure of the symptoms. The symptom class contains information

Patient_entry		
int	id	#parts number
str	user_name	#list of parts
int	sym_1	#if value is 3, means "asked yes" #if value is 4, means "asked no"
int	sym_2	
int	sym_3	142 symptoms here
int	.	
int	.	
int	.	
str	illness	#the illness name in the dataset
str	therapy	#the therapy number combine with illness
int	report	#Whether to generate a report, 1 yes, 0 no
int	owner	#who owner this entry, core process or um process

Figure 6: data structure in the database

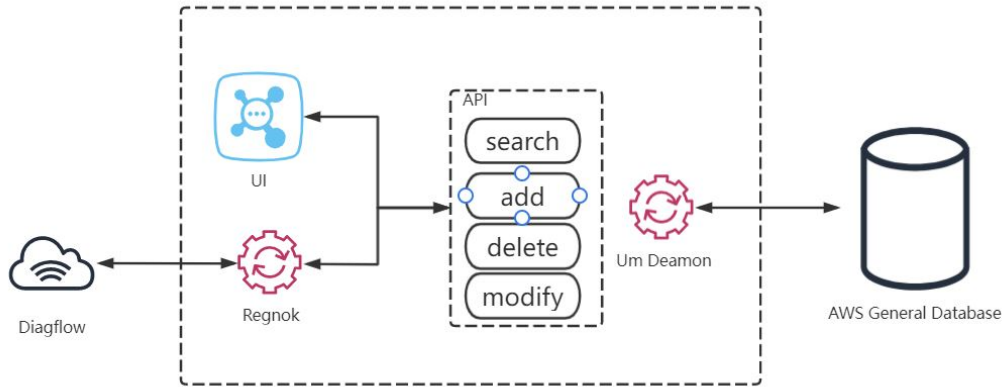


Figure 7: Work flow

such as the number of parts, the name of the part, the description of the symptom, etc. The task of our chatbot is to compose this information into a sentence to express it, and here we use the simplest method – a template to generate the sentence. For example, "eye\_pain" is a symptom name in our dataset. First we divide it into parts "eye" and description "pain", there are 1 part and 1 description so this example match the "1 part 1 description" template – "do you have description in part", the reply will be "Do you have pain in eyes?".

Now we just have a simple division of these symptoms, just to meet the needs of MVP. If we want this chatbot to be more intelligent, we can add more attributes to divide these symptoms and add more templates to match these symptoms.

## 5.4 Conclusion

We've successfully implemented a chatbot which is able to gather a patients symptoms, intelligently generate questions and provide a potential diagnosis to a doctor. We've had to overcome some challenges in order to do it and with more time we hope to further improve our product and approach clinics about conducting a trial run for the service. With feedback from these implementations we can get DASH ready for market and ready to help our front line medical workers.

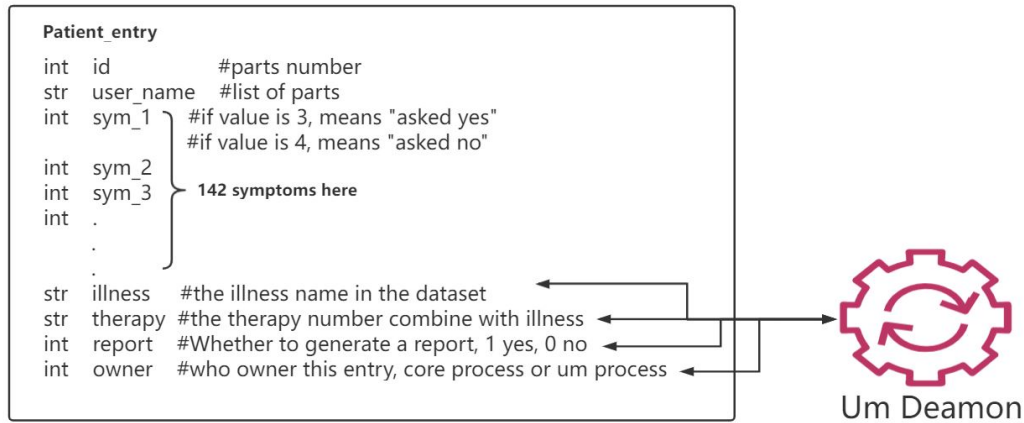


Figure 8: UM deamon

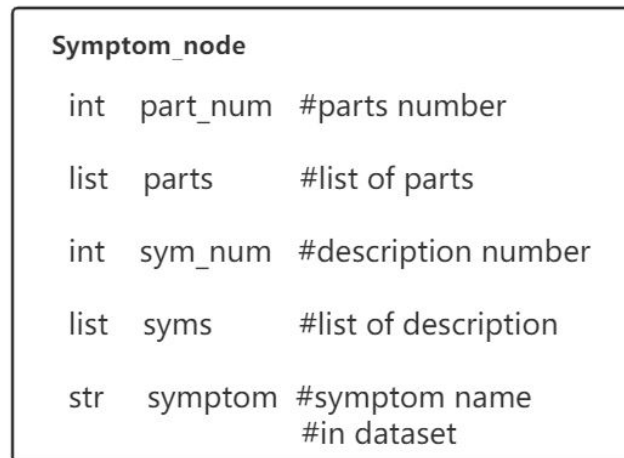


Figure 9: symptom node

#### 5.4.1 Future Development

The system provide as part of this project serves as an MVP, in order to get the product in the hand of users early and gather feedback. We see the potential for many further enhancements to the product over time:

- **Prescription recommendation and interaction analysis**  
Recommend a prescription for the patients diagnosis and take into account the potential drug interactions for prescription they are already taking.
- **Including the patients medical history in the model**  
If we have access to the patients medical history we can use this to better diagnose their symptoms.
- **Expanding the possible diseases with a larger dataset**  
Increasing the database to include more possible diagnoses to cover a greater number of illnesses.
- **Taking in doctor's feedback to improve the accuracy of the model**  
Using doctor's feedback to confirm later if the original diagnosis from the system was correct and if it was not using this information to improve the accuracy of the model.

- **Using phone sensors or smart devices to measure information about the patient**

With additional information coming from the sensor data we could expand our capabilities to take in blood pressure, heart rate or temperature into the diagnosis process. Additionally we could expand the system to continuously monitor the patient in their day to day for any anomalies.

## References

- [CCH19] Ching-Chin Chern, Yu-Jen Chen, and Bo Hsiao. Decision tree-based classifier in providing telehealth service. *BMC medical informatics and decision making*, 19(1):1–15, 2019.
- [ESG16] I Ketut Agung Enriko, Muhammad Suryanegara, and Dadang Gunawan. Heart disease prediction system using k-nearest neighbor algorithm with simplified patient’s health parameters. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 8(12):59–65, 2016.
- [Fut17] The Medical Futurist. The top 12 health chatbots. *The Medical Futurist*, 2017.
- [Kwo21] Liz Kwo. Contributed: Top 10 use cases for ai in healthcare. *mobihealthnews*, 2021.
- [oH21] Ministry of Health. Artificial intelligence in healthcare. *MOH*, 2021.
- [RD21] Reports and Data. Artificial intelligence (ai) in healthcare market size worth \$61.59 billion by 2027 — cagr of 43.6%. *globenewswire*, 2021.
- [XB19] Wenchao Xing and Yilin Bei. Medical health big data classification based on knn classification algorithm. *IEEE Access*, 8:28808–28819, 2019.

## A APPENDIX A - Project Proposal

## PROJECT PROPOSAL

Date of proposal: 28/10/2021
Project Title: Medical chatbot
Group ID (As Enrolled in LumiNUS Class Groups):  Group Members (Name , Student ID): Yang Yizhou           A0231320A Wang Hongtao        A0195158U <b>Tadhg Kennedy</b> A0231552N Meng Chenxi          A0231546J
Sponsor/Client: <i>(Company Name, Address and Contact Name, Email, if any)</i>  <i>None</i>
Background/Aims/Objectives:  <p>Initially we come up with creating a medical chatbot because nowadays we are facing the situation that patients are hard to know what diseases that they are facing so that they also cannot register the correct department in the hospital.</p> <p>In order to solve the problem we come up with the idea that creating a medical chatbot to make people easier to know their illness by answering the symptoms questions.</p> <p>Also the COVID-19 pandemic has changed the way we approach doctor's visits and medical institution as a whole. With a highly transmissible disease out in the population having sick people congregate in a location is the exact opposite of what doctors and immunologists advise. While the COVID-19 pandemic is waning the impacts it has had on our approach to medicine can be seen across the industry, tele-medicine services are now common place and efforts to reduce patient congregation are still in place to reduce the chances of a potential spike.</p> <p>Our medical chatbot system seeks to address this need and provide additional benefits to the institutions it's deployed in. By leveraging the power of machine learning and chatbots we hope to be able to reduce the need for patients to congregate in waiting rooms and reduce some of the workload from the doctors in their already busy workdays. Our system will collect the patient symptoms via a chatbot, these symptoms will then be used to intelligently generate questions to help narrow down the possible diagnoses, once the system has sufficient information the symptoms will be used to provide a potential diagnosis in a report to the doctor. This report will allow the doctor to decide if the patient</p>



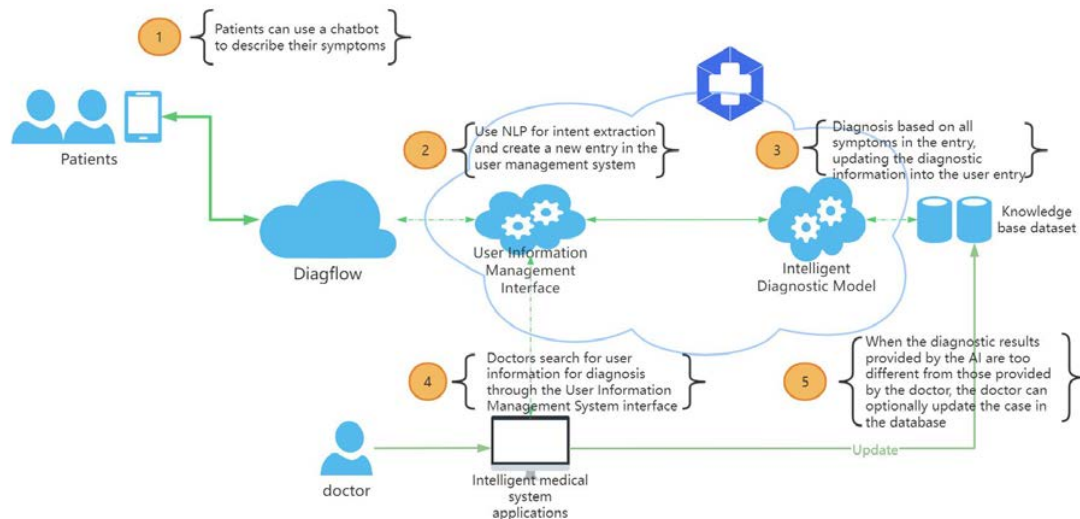
requires an in person consultation, tele-medicine session, provide prescription or potentially skip the visit to the GP and proceed straight to a specialist.

### Project Descriptions:

We can divide our medical chatbot into following parts:

1. Train the model using decision tree and KNN. Dataset is based on the symptoms and illness of online dataset.
2. Use chatbot to collect the symptoms from the patients.
3. Use the model to predict the illness based on the symptoms that chatbot collected and confirmed.
4. For doctors we create a chatbot which can display the result of patients information, symptoms and illness.

The flowchart:



## B APPENDIX B

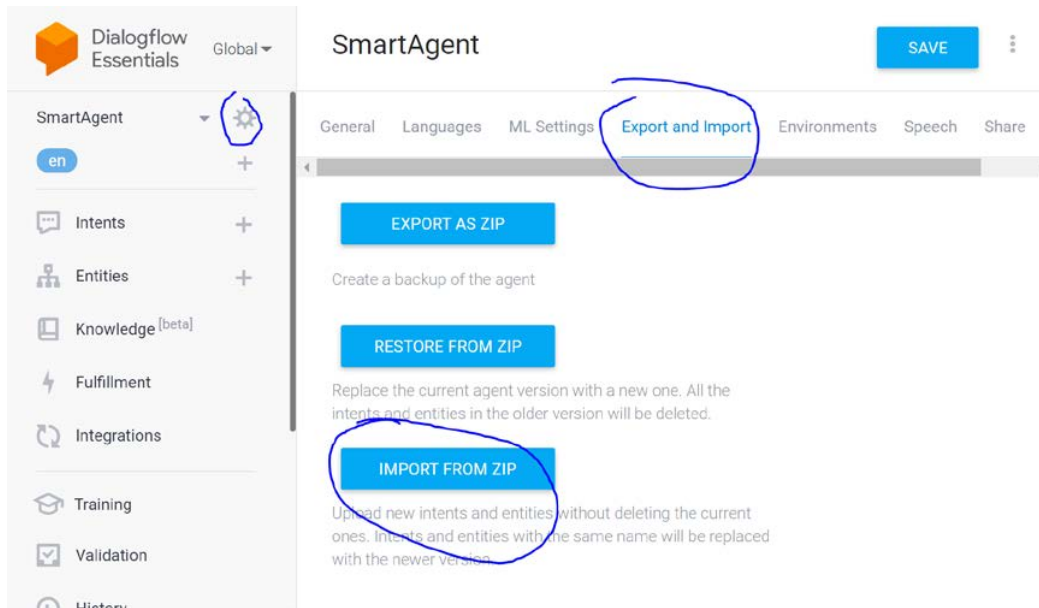
Mapped System Functionalities against knowledge, techniques and skills of module courses

Module Courses	System Functionalities/Techniques Applied
Machine Reasoning(MR)	<b>Knowledge Representation</b> <b>Supervised Learning Algorithms(KNN and decision tree)</b> <b>Rule Based System</b>
Reasoning Systems(RS)	<b>Analytic and Synthetic tasks</b>
Cognitive System(CGS)	<b>NLP</b> <b>Chatbot</b> <b>diagflow</b>

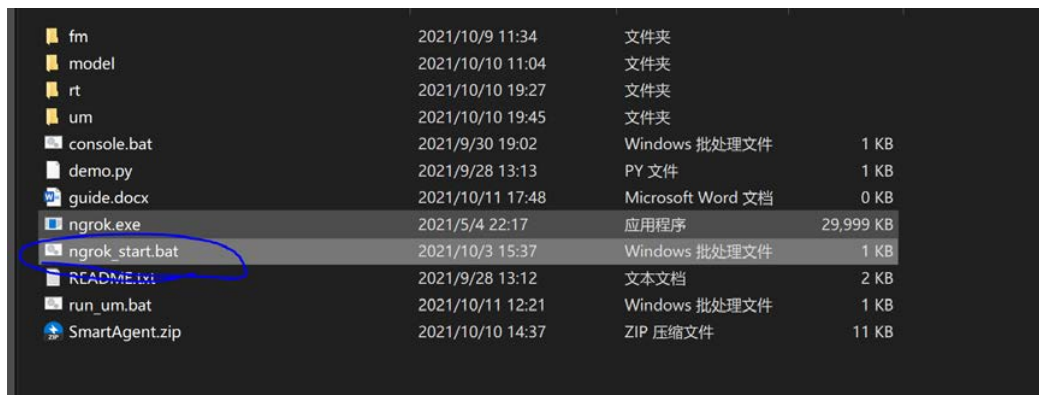
## C APPENDIX C - Installation & User Guide

Step 1: login your diagflow dashboard

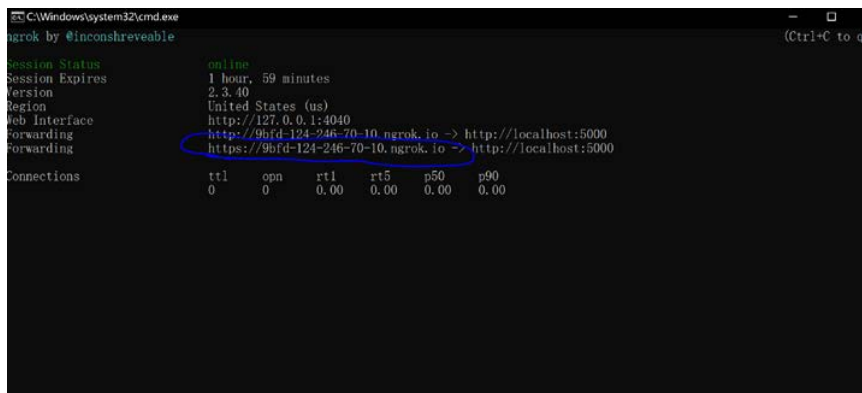
Step 2: import SmartAgent.zip



Step 3: click ngrok\_start.bat



Step 4: copy to fulfillment



## Webhook


ENABLED 

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.


URL\*

BASIC AUTH

HEADERS

 Add header

SMALL TALK

 Dialogflow Essentials

Global

SmartAgent

en

Intents

Entities

Knowledge [beta]

**Fulfillment**


Integrations

Training

Validation

History

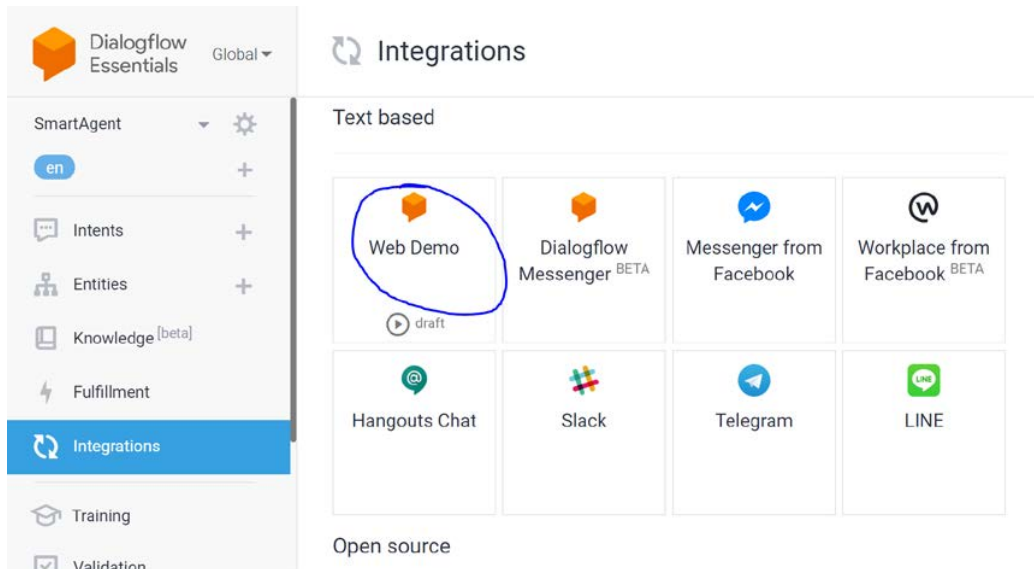
Analytics

 Fulfillment

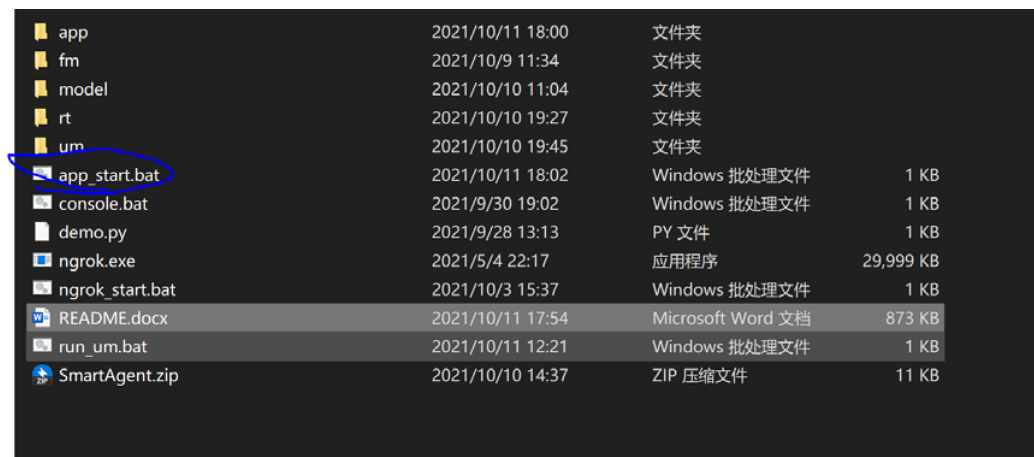
index.js package.json

```
1 // See https://github.com/dialogflow/dialogflow-fulfillment-nodejs
2 // for Dialogflow fulfillment library docs, samples, and to report issues
3 'use strict';
4
5 const functions = require('firebase-functions');
6 const WebhookClient = require('dialogflow-fulfillment');
7 const {Card, Suggestion} = require('dialogflow-fulfillment');
8
9 process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements
10
11 exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
12   const agent = new WebhookClient({ request, response });
13   console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
14   console.log('Dialogflow Request body: ' + JSON.stringify(request.body));
15
16   // ...
17
18   // ...
19
20   // ...
21
22   // ...
23
24   // ...
25
26   // ...
27
28   // ...
29
30   // ...
31
32   // ...
33
34   // ...
35
36   // ...
37
38   // ...
39
40   // ...
41
42   // ...
43
44   // ...
45
46   // ...
47
48   // ...
49
50   // ...
51
52   // ...
53
54   // ...
55
56   // ...
57
58   // ...
59
60   // ...
61
62   // ...
63
64   // ...
65
66   // ...
67
68   // ...
69
70   // ...
71
72   // ...
73
74   // ...
75
76   // ...
77
78   // ...
79
80   // ...
81
82   // ...
83
84   // ...
85
86   // ...
87
88   // ...
89
90   // ...
91
92   // ...
93
94   // ...
95
96   // ...
97
98   // ...
99
100  // ...
101
102  // ...
103
104  // ...
105
106  // ...
107
108  // ...
109
110  // ...
111
112  // ...
113
114  // ...
115
116  // ...
117
118  // ...
119
120  // ...
121
122  // ...
123
124  // ...
125
126  // ...
127
128  // ...
129
130  // ...
131
132  // ...
133
134  // ...
135
136  // ...
137
138  // ...
139
140  // ...
141
142  // ...
143
144  // ...
145
146  // ...
147
148  // ...
149
150  // ...
151
152  // ...
153
154  // ...
155
156  // ...
157
158  // ...
159
160  // ...
161
162  // ...
163
164  // ...
165
166  // ...
167
168  // ...
169
170  // ...
171
172  // ...
173
174  // ...
175
176  // ...
177
178  // ...
179
180  // ...
181
182  // ...
183
184  // ...
185
186  // ...
187
188  // ...
189
190  // ...
191
192  // ...
193
194  // ...
195
196  // ...
197
198  // ...
199
200  // ...
201
202  // ...
203
204  // ...
205
206  // ...
207
208  // ...
209
210  // ...
211
212  // ...
213
214  // ...
215
216  // ...
217
218  // ...
219
220  // ...
221
222  // ...
223
224  // ...
225
226  // ...
227
228  // ...
229
230  // ...
231
232  // ...
233
234  // ...
235
236  // ...
237
238  // ...
239
240  // ...
241
242  // ...
243
244  // ...
245
246  // ...
247
248  // ...
249
250  // ...
251
252  // ...
253
254  // ...
255
256  // ...
257
258  // ...
259
260  // ...
261
262  // ...
263
264  // ...
265
266  // ...
267
268  // ...
269
270  // ...
271
272  // ...
273
274  // ...
275
276  // ...
277
278  // ...
279
280  // ...
281
282  // ...
283
284  // ...
285
286  // ...
287
288  // ...
289
290  // ...
291
292  // ...
293
294  // ...
295
296  // ...
297
298  // ...
299
300  // ...
301
302  // ...
303
304  // ...
305
306  // ...
307
308  // ...
309
310  // ...
311
312  // ...
313
314  // ...
315
316  // ...
317
318  // ...
319
320  // ...
321
322  // ...
323
324  // ...
325
326  // ...
327
328  // ...
329
330  // ...
331
332  // ...
333
334  // ...
335
336  // ...
337
338  // ...
339
340  // ...
341
342  // ...
343
344  // ...
345
346  // ...
347
348  // ...
349
350  // ...
351
352  // ...
353
354  // ...
355
356  // ...
357
358  // ...
359
360  // ...
361
362  // ...
363
364  // ...
365
366  // ...
367
368  // ...
369
370  // ...
371
372  // ...
373
374  // ...
375
376  // ...
377
378  // ...
379
380  // ...
381
382  // ...
383
384  // ...
385
386  // ...
387
388  // ...
389
390  // ...
391
392  // ...
393
394  // ...
395
396  // ...
397
398  // ...
399
400  // ...
401
402  // ...
403
404  // ...
405
406  // ...
407
408  // ...
409
410  // ...
411
412  // ...
413
414  // ...
415
416  // ...
417
418  // ...
419
420  // ...
421
422  // ...
423
424  // ...
425
426  // ...
427
428  // ...
429
430  // ...
431
432  // ...
433
434  // ...
435
436  // ...
437
438  // ...
439
440  // ...
441
442  // ...
443
444  // ...
445
446  // ...
447
448  // ...
449
450  // ...
451
452  // ...
453
454  // ...
455
456  // ...
457
458  // ...
459
460  // ...
461
462  // ...
463
464  // ...
465
466  // ...
467
468  // ...
469
470  // ...
471
472  // ...
473
474  // ...
475
476  // ...
477
478  // ...
479
480  // ...
481
482  // ...
483
484  // ...
485
486  // ...
487
488  // ...
489
490  // ...
491
492  // ...
493
494  // ...
495
496  // ...
497
498  // ...
499
500  // ...
501
502  // ...
503
504  // ...
505
506  // ...
507
508  // ...
509
510  // ...
511
512  // ...
513
514  // ...
515
516  // ...
517
518  // ...
519
520  // ...
521
522  // ...
523
524  // ...
525
526  // ...
527
528  // ...
529
530  // ...
531
532  // ...
533
534  // ...
535
536  // ...
537
538  // ...
539
540  // ...
541
542  // ...
543
544  // ...
545
546  // ...
547
548  // ...
549
550  // ...
551
552  // ...
553
554  // ...
555
556  // ...
557
558  // ...
559
560  // ...
561
562  // ...
563
564  // ...
565
566  // ...
567
568  // ...
569
570  // ...
571
572  // ...
573
574  // ...
575
576  // ...
577
578  // ...
579
580  // ...
581
582  // ...
583
584  // ...
585
586  // ...
587
588  // ...
589
590  // ...
591
592  // ...
593
594  // ...
595
596  // ...
597
598  // ...
599
600  // ...
601
602  // ...
603
604  // ...
605
606  // ...
607
608  // ...
609
610  // ...
611
612  // ...
613
614  // ...
615
616  // ...
617
618  // ...
619
620  // ...
621
622  // ...
623
624  // ...
625
626  // ...
627
628  // ...
629
630  // ...
631
632  // ...
633
634  // ...
635
636  // ...
637
638  // ...
639
640  // ...
641
642  // ...
643
644  // ...
645
646  // ...
647
648  // ...
649
650  // ...
651
652  // ...
653
654  // ...
655
656  // ...
657
658  // ...
659
660  // ...
661
662  // ...
663
664  // ...
665
666  // ...
667
668  // ...
669
670  // ...
671
672  // ...
673
674  // ...
675
676  // ...
677
678  // ...
679
680  // ...
681
682  // ...
683
684  // ...
685
686  // ...
687
688  // ...
689
690  // ...
691
692  // ...
693
694  // ...
695
696  // ...
697
698  // ...
699
700  // ...
701
702  // ...
703
704  // ...
705
706  // ...
707
708  // ...
709
710  // ...
711
712  // ...
713
714  // ...
715
716  // ...
717
718  // ...
719
720  // ...
721
722  // ...
723
724  // ...
725
726  // ...
727
728  // ...
729
730  // ...
731
732  // ...
733
734  // ...
735
736  // ...
737
738  // ...
739
740  // ...
741
742  // ...
743
744  // ...
745
746  // ...
747
748  // ...
749
750  // ...
751
752  // ...
753
754  // ...
755
756  // ...
757
758  // ...
759
760  // ...
761
762  // ...
763
764  // ...
765
766  // ...
767
768  // ...
769
770  // ...
771
772  // ...
773
774  // ...
775
776  // ...
777
778  // ...
779
780  // ...
781
782  // ...
783
784  // ...
785
786  // ...
787
788  // ...
789
790  // ...
791
792  // ...
793
794  // ...
795
796  // ...
797
798  // ...
799
800  // ...
801
802  // ...
803
804  // ...
805
806  // ...
807
808  // ...
809
810  // ...
811
812  // ...
813
814  // ...
815
816  // ...
817
818  // ...
819
820  // ...
821
822  // ...
823
824  // ...
825
826  // ...
827
828  // ...
829
830  // ...
831
832  // ...
833
834  // ...
835
836  // ...
837
838  // ...
839
840  // ...
841
842  // ...
843
844  // ...
845
846  // ...
847
848  // ...
849
850  // ...
851
852  // ...
853
854  // ...
855
856  // ...
857
858  // ...
859
860  // ...
861
862  // ...
863
864  // ...
865
866  // ...
867
868  // ...
869
870  // ...
871
872  // ...
873
874  // ...
875
876  // ...
877
878  // ...
879
880  // ...
881
882  // ...
883
884  // ...
885
886  // ...
887
888  // ...
889
890  // ...
891
892  // ...
893
894  // ...
895
896  // ...
897
898  // ...
899
900  // ...
901
902  // ...
903
904  // ...
905
906  // ...
907
908  // ...
909
910  // ...
911
912  // ...
913
914  // ...
915
916  // ...
917
918  // ...
919
920  // ...
921
922  // ...
923
924  // ...
925
926  // ...
927
928  // ...
929
930  // ...
931
932  // ...
933
934  // ...
935
936  // ...
937
938  // ...
939
940  // ...
941
942  // ...
943
944  // ...
945
946  // ...
947
948  // ...
949
950  // ...
951
952  // ...
953
954  // ...
955
956  // ...
957
958  // ...
959
960  // ...
961
962  // ...
963
964  // ...
965
966  // ...
967
968  // ...
969
970  // ...
971
972  // ...
973
974  // ...
975
976  // ...
977
978  // ...
979
980  // ...
981
982  // ...
983
984  // ...
985
986  // ...
987
988  // ...
989
990  // ...
991
992  // ...
993
994  // ...
995
996  // ...
997
998  // ...
999
1000  // ...
1001
1002  // ...
1003
1004  // ...
1005
1006  // ...
1007
1008  // ...
1009
1010  // ...
1011
1012  // ...
1013
1014  // ...
1015
1016  // ...
1017
1018  // ...
1019
1020  // ...
1021
1022  // ...
1023
1024  // ...
1025
1026  // ...
1027
1028  // ...
1029
1030  // ...
1031
1032  // ...
1033
1034  // ...
1035
1036  // ...
1037
1038  // ...
1039
1040  // ...
1041
1042  // ...
1043
1044  // ...
1045
1046  // ...
1047
1048  // ...
1049
1050  // ...
1051
1052  // ...
1053
1054  // ...
1055
1056  // ...
1057
1058  // ...
1059
1060  // ...
1061
1062  // ...
1063
1064  // ...
1065
1066  // ...
1067
1068  // ...
1069
1070  // ...
1071
1072  // ...
1073
1074  // ...
1075
1076  // ...
1077
1078  // ...
1079
1080  // ...
1081
1082  // ...
1083
1084  // ...
1085
1086  // ...
1087
1088  // ...
1089
1090  // ...
1091
1092  // ...
1093
1094  // ...
1095
1096  // ...
1097
1098  // ...
1099
1100  // ...
1101
1102  // ...
1103
1104  // ...
1105
1106  // ...
1107
1108  // ...
1109
1110  // ...
1111
1112  // ...
1113
1114  // ...
1115
1116  // ...
1117
1118  // ...
1119
1120  // ...
1121
1122  // ...
1123
1124  // ...
1125
1126  // ...
1127
1128  // ...
1129
1130  // ...
1131
1132  // ...
1133
1134  // ...
1135
1136  // ...
1137
1138  // ...
1139
1140  // ...
1141
1142  // ...
1143
1144  // ...
1145
1146  // ...
1147
1148  // ...
1149
1150  // ...
1151
1152  // ...
1153
1154  // ...
1155
1156  // ...
1157
1158  // ...
1159
1160  // ...
1161
1162  // ...
1163
1164  // ...
1165
1166  // ...
1167
1168  // ...
1169
1170  // ...
1171
1172  // ...
1173
1174  // ...
1175
1176  // ...
1177
1178  // ...
1179
1180  // ...
1181
1182  // ...
1183
1184  // ...
1185
1186  // ...
1187
1188  // ...
1189
1190  // ...
1191
1192  // ...
1193
1194  // ...
1195
1196  // ...
1197
1198  // ...
1199
1200  // ...
1201
1202  // ...
1203
1204  // ...
1205
1206  // ...
1207
1208  // ...
1209
1210  // ...
1211
1212  // ...
1213
1214  // ...
1215
1216  // ...
1217
1218  // ...
1219
1220  // ...
1221
1222  // ...
1223
1224  // ...
1225
1226  // ...
1227
1228  // ...
1229
1230  // ...
1231
1232  // ...
1233
1234  // ...
1235
1236  // ...
1237
1238  // ...
1239
1240  // ...
1241
1242  // ...
1243
1244  // ...
1245
1246  // ...
1247
1248  // ...
1249
1250  // ...
1251
1252  // ...
1253
1254  // ...
1255
1256  // ...
1257
1258  // ...
1259
1260  // ...
1261
1262  // ...
1263
1264  // ...
1265
1266  // ...
1267
1268  // ...
1269
1270  // ...
1271
1272  // ...
1273
1274  // ...
1275
1276  // ...
1277
1278  // ...
1279
1280  // ...
1281
1282  // ...
1283
1284  // ...
1285
1286  // ...
1287
1288  // ...
1289
1290  // ...
1291
1292  // ...
1293
1294  // ...
1295
1296  // ...
1297
1298  // ...
1299
1300  // ...
1301
1302  // ...
1303
1304  // ...
1305
1306  // ...
1307
1308  // ...
1309
1310  // ...
1311
1312  // ...
1313
1314  // ...
1315
1316  // ...
1317
1318  // ...
1319
1320  // ...
1321
1322  // ...
1323
1324  // ...
1325
1326  // ...
1327
1328  // ...
1329
1330  // ...
1331
1332  // ...
1333
1334  // ...
1335
1336  // ...
1337
1338  // ...
1339
1340  // ...
1341
1342  // ...
1343
1344  // ...
1345
1346  // ...
1347
1348  // ...
1349
1350  // ...
1351
1352  // ...
1353
1354  // ...
1355
1356  // ...
1357
1358  // ...
1359
1360  // ...
1361
1362  // ...
1363
1364  // ...
1365
1366  // ...
1367
1368  // ...
1369
1370  // ...
1371
1372  // ...
1373
1374  // ...
1375
1376  // ...
1377
1378  // ...
1379
1380  // ...
1381
1382  // ...
1383
1384  // ...
1385
1386  // ...
1387
1388  // ...
1389
1390  // ...
1391
1392  // ...
1393
1394  // ...
1395
1396  // ...
1397
1398  // ...
1399
1400  // ...
1401
1402  // ...
1403
1404  // ...
1405
1406  // ...
1407
1408  // ...
1409
1410  // ...
1411
1412  // ...
1413
1414  // ...
1415
1416  // ...
1417
1418  // ...
1419
1420  // ...
1421
1422  // ...
1423
1424  // ...
1425
1426  // ...
1427
1428  // ...
1429
1430  // ...
1431
1432  // ...
1433
1434  // ...
1435
1436  // ...
1437
1438  // ...
1439
1440  // ...
1441
1442  // ...
1443
1444  // ...
1445
1446  // ...
1447
1448  // ...
1449
1450  // ...
1451
1452  // ...
1453
1454  // ...
1455
1456  // ...
1457
1458  // ...
1459
1460  // ...
1461
1462  // ...
1463
1464  // ...
1465
1466  // ...
1467
1468  // ...
1469
1470  // ...
1471
1472  // ...
1473
1474  // ...
1475
1476  // ...
1477
1478  // ...
1479
1480  // ...
1481
1482  // ...
1483
1484  // ...
1485
1486  // ...
1487
1488  // ...
1489
1490  // ...
1491
1492  // ...
1493
1494  // ...
1495
1496  // ...
1497
1498  // ...
1499
1500  // ...
1501
1502  // ...
1503
1504  // ...
1505
1506  // ...
1507
1508  // ...
1509
1510  // ...
1511
1512  // ...
1513
1514  // ...
1515
1516  // ...
1517
1518  // ...
1519
1520  // ...
1521
1522  // ...
1523
1524  // ...
1525
1526  // ...
1527
1528  // ...
1529
1530  // ...
1531
1532  // ...
1533
1534  // ...
1535
1536  // ...
1537
1538  // ...
1539
1540  // ...
1541
1542  // ...
1543
1544  // ...
1545
1546  // ...
1547
1548  // ...
1549
1550  // ...
1551
1552  // ...
1553
1554  // ...
1555
1556  // ...
1557
1558  // ...
1559
1560  // ...
1561
1562  // ...
1563
1564  // ...
1565
1566  // ...
1567
1568  // ...
1569
1570  // ...
1571
1572  // ...
1573
1574  // ...
1575
1576  // ...
1577
1578  // ...
1579
1580  // ...
1581
1582  // ...
1583
1584  // ...
1585
1586  // ...
1587
1588  // ...
1589
1590  // ...
1591
1592  // ...
1593
1594  // ...
1595
1596  // ...
1597
1598  // ...
1599
1600  // ...
1601
1602  // ...
1603
1604  // ...
1605
1606  // ...
1607
1608  // ...
1609
1610  // ...
1611
1612  // ...
1613
1614  // ...
1615
1616  // ...
1617
1618  // ...
1619
1620  // ...
1621
1622  // ...
1623
1624  // ...
1625
1626  // ...
1627
1628  // ...
1629
1630  // ...
1631
1632  // ...
1633
1634  // ...
1635
1636  // ...
1637
1638  // ...
1639
1640  // ...
1641
1642  // ...
1643
1644  // ...
1645
1646  // ...
1647
1648  // ...
1649
1650  // ...
1651
1652  // ...
1653
1654  // ...
1655
1656  // ...
1657
1658  // ...
1659
1660  // ...
1661
1662  // ...
1663
1664  // ...
1665
1666  // ...
1667
1668  // ...
1669
1670  // ...
1671
1672  // ...
1673
1674  // ...
1675
1676  // ...
1677
1678  // ...
1679
1680  // ...
1681
1682  // ...
1683
1684  // ...
1685
1686  // ...
1687
1688  // ...
1689
1690  // ...
1691
1692  // ...
1693
1694  // ...
1695
1696  // ...
1697
1698  // ...
1699
1700  // ...
1701
1702  // ...
1703
1704  // ...
1705
1706  // ...
1707
1708  // ...
1709
1710  // ...
1711
1712  // ...
1713
1714  // ...
1715
1716  // ...
1717
1718  // ...
1719
1720  // ...
1721
1722  // ...
1723
1724  // ...
1725
1726  // ...
1727
1728  // ...
1729
1730  // ...
1731
1732  // ...
1733
1734  // ...
1735
1736  // ...
1737
1738  // ...
1739
1740  // ...
1741
1742  // ...
1743
1744  // ...
1745
1746  // ...
1747
1748  // ...
1749
1750  // ...
1751
1752  // ...
1753
1754  // ...
1755
1756  // ...
1757
1758  // ...
1759
1760  // ...
1761
1762  // ...
1763
1764  // ...
1765
1766  // ...
1767
1768  // ...
1769
1770  // ...
1771
1772  // ...
1773
1774  // ...
1775
1776  // ...
1777
1778  // ...
1779
1780  // ...
1781
1782  // ...
1783
1784  // ...
1785
1786  // ...
1787
1788  // ...
1789
1790  // ...
1791
1792  // ...
1793
1794  // ...
1795
1796  // ...
1797
1798  // ...
1799
1800  // ...
1801
1802  // ...
1803
1804  // ...
1805
1806  // ...
1807
1808  // ...
1809
1810  // ...
1811
1812  // ...
1813
1814  // ...
1815
1816  // ...
1817
1818  // ...
1819
1820  // ...
1821
1822  // ...
1823
1824  // ...
1825
1826  // ...
1827
1828  // ...
1829
1830  // ...
1831
1832  // ...
1833
1834  // ...
1835
1836  // ...
1837
1838  // ...
1839
1840  // ...
1841
1842  // ...
1843
1844  // ...
1845
1846  // ...
1847
1848  // ...
1849
1850  // ...
1851
1852  // ...
1853
1854  // ...
1855
1856  // ...
1857
1858  // ...
1859
1860  // ...
1861
1862  // ...
1863
1864  // ...
1865
1866  // ...
1867
1868  // ...
1869
1870  // ...
1871
1872  // ...
1873
1874  // ...
1875
1876  // ...
1877
1878  // ...
1879
1880  // ...
1881
1882  // ...
1883
1884  // ...
1885
1886  // ...
1887
1888  // ...
1889
1890  // ...
1891
1892  // ...
1893
1894  // ...
1895
1896  // ...
1897
1898  // ...
1899
1900  // ...
1901
1902  // ...
1903
1904  // ...
1905
1906  // ...
1907
1908  // ...
1909
1910  // ...
1911
1912  // ...
1913
1914  // ...
1915
1916  // ...
1917
1918  // ...
1919
1920  // ...
1921
1922  // ...
1923
1924  // ...
1925
1926  // ...
1927
1928  // ...
1929
1930  // ...
1931
1932  // ...
1933
1934  // ...
1935
1936  // ...
1937
1938  // ...
1939
1940  // ...
1941
1942  // ...
1943
1944  // ...
1945
1946  // ...
1947
1948  // ...
1949
1950  // ...
1951
1952  // ...
1953
1954  // ...
1955
1956  // ...
1957
1958  // ...
1959
1960  // ...
1961
1962  // ...
1963
1964  // ...
1965
1966  // ...
1967
1
```

Step 6: have a try



Step 7: click app\_start.bat to start app



Referring to the following GITHUB link: [https://github.com/wanghongtaonus/IRS\\_PM\\_2021\\_07\\_05\\_IS03FT\\_GRP\\_17\\_DASH/blob/main/Miscellaneous/Installation\\_guide.docx](https://github.com/wanghongtaonus/IRS_PM_2021_07_05_IS03FT_GRP_17_DASH/blob/main/Miscellaneous/Installation_guide.docx)

## D APPENDIX D - Doctor Interview

Second year family medicine resident at Geylang Polyclinic.

**How many patient's would you see typically in a day at the poly clinic?**

It can range from around 20 to low 30s, on average about 25 per day. This is slightly lower than usual because the clinic recently changed the computer operating system and in order to ensure time to properly record everything the number of slots has been reduced to 4 per hour, where previously it was 6.

**Is this pretty typical for all clinics or are there more high traffic locations?**

Generally, pretty similar across the country.

**How many patients would you say come in with ailments that don't require a prescription? Could be treated with bedrest or painkillers?**

It's difficult to say, because different rooms are assigned different types of cases. Some will be allocated chronic cases with follow-ups, and some will be assigned acute cases. But in my experience for acute cases this can be as high as 60

**How much time do you have with each patient?**

This differs between chronic and acute cases, for acute cases about 10 minutes is allocated.

**Do you have a set list of questions you ask or is it more by intuition?**

It's a mix, we're taught a set of question you should ask but based on how the patient looks and how they answer questions it changes. Every doctor has a list in their head to rule out or in illnesses.

**Have you ever used any kind of AI in your work?**

Never, it's rarely mentioned in teaching either. It may be a case where schools want to ensure doctors have the full set of skills and don't rely on AI as a crutch/become dependent on it.

**More specifically have you ever used or see any diagnosis aids using AI?**

Question is moot from above answer.

**Do you feel like something which gathered patients symptoms in advance and gave you a potential diagnosis could help you in our day to day work?**

I feel like it could be useful in helping reduce the number of patients. I would, however, be concerned that the model's diagnosis could lead to bias in the diagnosis with the doctors. I've have seen instances where going into a consultation with a preconceived diagnosis can impact the questions asked and how the patients answers were interpreted to fit idea.

## E APPENDIX E - Individual Report

## **E.1 Personal Report: Tadhg Kennedy**

### **E.1.1 Personal Contribution**

Based on my previous experience working in software and product development, I feel I brought some important insight into the importance for ensuring everyone is on the same page and working towards the same overall goal. I worked to make sure we discussed our ideas early so we could refine the scope to a workable project plan.

My experience with development and understanding the importance of testing practices meant that we were able to identify critical bugs in the system.

I took on the role of administrative leader during the course of the project, using the skills I previously developed to work with the team to set up meetings for in person discussion (when restrictions allowed) and keeping the team up to date on progress. I worked with the team in these meetings to identify the next steps we needed to take and divide out the work between everyone with clear deadlines or checkpoints to make sure everyone was progressing and didn't have any roadblocks which needed help.

### **E.1.2 Lessons Learned**

Through discussion and working with my team I learned a lot about the different models we investigated using and was introduced to functions I hadn't previously used in the course studies so far.

I also learned about the different medical care structure in China, and how it differed from Singapore during our discussion about the project when my team mates didn't know about the GP clinics in Singapore and the importance of a referral letter. This lead to important discussions about initial target market for the product, as the different structures would require different approaches.

Coming from a professional environment with large teams working together it took me a while to understand that it was important to keep scope manageable for a smaller team, especially when the group was split between different class schedules and work for other projects.

I also learned the importance of speaking clearly and not falling into old habits using terminology or acronyms from my previous employment which weren't commonly used elsewhere. I can naturally fall into speaking very quickly and with my accent this can be difficult for those not familiar with it.

### **E.1.3 Future**

I'm sure the knowledge I've gained about models such as decision trees and kNN will be useful to me going forward in my ML career.

Additionally, I think that what I've learned from working cross culturally with my team mates will better prepare me for working with a greater diversity of teams and team members.

## **E.2 Personal Report: Wang Hongtao**

### **E.2.1 Personal Contribution**

During this project, I was involved the whole process of the system design and integration including use case design and development, back end function and UI design and development.

### **E.2.2 Lessons Learned**

I learn a lot from this project, firstly we collect suggestion from domain expert, so we build a survey to do marketing in order to know requirements of patients and doctor, and i think it is essential because pain point is the key for your product and grabbing the customer.

In addition, Our team find dataset online and try different models such as decision tree and knn, we evaluate the performance and choose the better, sometimes need trade-off.

The main part for me is Ui design and development, i learn pyqt and put it into practice.

One of the interesting part in this project is during the discussion i know the differences of medical system between different regions, so that i know when solving problem we need to treat problem case by case.

And Group members are from different regions, so it help a lot for collaborating with diverse team members.

My experience with development and understanding the importance of testing practices meant that we were able to identify critical bugs in the system.

### **E.2.3 Future**

After this intelligent reasoning system practice module, I understood the whole process of system design and development which can help me a lot in the working field, from gathering requirements, use case design and development, group discussion...

I also investigate different models, these practice will help me better understand the concepts which i am sure will useful for me.



## **E.3 Personal Report: Meng Chenxi**

### **E.3.1 Personal Contribution**

At the beginning of IRS module, I gathered this team considering everyone's excellent capability and working experience. As a fresh graduate, I don't have much working experience, which means I always performed a learner and executor working with other team members during the whole project.

I sometimes did minutes of the project meeting. Mostly, I did the system modeling. After investigation of some relevant papers and projects, I build KNN model and decision tree model, which is a very important part in this system and I also did model testing, which contained parameters adjusting and debugging, in order to optimize the system efficiency. What's more, I made a demo of how the dialogflow use, which was performed in our commercial video.

### **E.3.2 Lessons Learned**

I totally involved in this project using my abilities based on previous experience and passion to complete this useful system. As a graduate of statistics, I am quite unfamiliar with python and how to build a machine learning system. During this project, I learned some practical techniques of building machine learning models by actual operations. In addition, I learned how to organize a whole system and how to combine chatbot with model from my teammates while having meeting with them.

### **E.3.3 Future**

I am so glad to see our system work well and solve real world problems, which inspires me to go deeper into artificial intelligent area. I will learn more knowledge and skills about modeling and system integration, which are very helpful in my future work. I also feel very lucky and grateful of having those wonderful teammates who help me a lot and show kind patience. I am sure this project experience will have great influence on myself and my future.

## **E.4 Personal Report: Yang Yizhou**

### **E.4.1 Personal Contribution**

Propose a Chatbot solution: For the whole solution, I propose to extract the intent in the sentences through Dialogflow and design a whole backend framework to handle these intents and reply based on templates to implement a simple bot. I also propose to use KNN model as an inference model, because KNN can predict categories based on distance. The main feature of a disease is the symptoms, and KNN can suitably determine which disease a collection of symptoms belongs to, while KNN is also well suited for expandable and continuously updated databases.

Backend framework & database design and implementation: I designed a low-coupling and high-cohesion backend framework, where modules and modules are separated from each other, making it possible to do the coding together. At the same time, this backend framework is also highly extensible, as long as different functions are added to the functional modules, then it makes the system handle different situations. The framework design of the database is inspired by the registers in the embedded system, although I don't have much is database experience, but I have a lot of experience in embedded development, I know how to design and use the registers. If you think of the database as a hardware device, then each entry is a register in it, and the background program is the firmware of this device, so I designed such a structure.

### **E.4.2 Lessons Learned**

I don't actually have any experience designing and implementing a system in Python. But it's not that hard for me to build a simple system with some knowledge of locks, inter-thread communication, and inter-process communication implemented in Python. In terms of machine learning, this project was very rewarding for me, and we did a lot of research during the initial discussions of the proposal. The main problem was how to find the next problem when we know some of the symptoms. In this process, we investigated KNN model and D-tree, and found that KNN model is better than D-tree in terms of data set and interpretability, so we chose KNN as the inference model. We also took a lot of detours in the later implementation process, but this detour also allowed us to learn a lot of machine learning engineering stuff. For example, the accuracy of our model is high but the next problem we get in chatbot is often unsatisfactory. Finally, we found out that the metric function was wrong, because our data is a binary vector, so using Euclidean distance to calculate the distance would be very poor, but if we use "russellrao" to calculate the distance, the performance is much better. In summary, this project has given me the ability to translate machine learning theory into practical engineering, and has increased my experience in machine learning engineering.

### **E.4.3 Future**

This project has given me engineering experience in NLP. I got familiar with how to use Dialogflow to deploy projects and publish my chatbot. I also learned how to use ngrok to make the local computer a server to handle diagflow requests. During model deployment, I also learned how to use sklearn library to train and deploy KNN models. Also, I had a lot of debug experience in this project. In addition to the technical aspects, I also learned a lot about the market, how to research user needs. How to analyze the market situation, and so on. All in all, with the cooperation between me and my team members, we were able to complete this relatively complex project. And everyone delivered their own module as module leader and learned something new, which I believe is meaningful for every member of the group.