SIA ASSIST
PERSONAL AIRPORT ASSISTANT

COGNITIVE SYSTEMS PROJECT

GROUP MEMBERS	*STUDENT ID*
Aniket Mohan Arasanipalai	A0198481N
Srikar Namburu	A0198490N
Vishaag Suriyanarayanan	A0198504X
Akshay Sachdeva	A0198532W

Contents

EXECUTIVE SUMMARY

Artificial intelligent chatbots are becoming more and more prevalent form of service technology in a variety of scenarios in industries ranging from banking, customer services, logistics and education. Traditional computer interfaces require structured and predictable input in order to function properly, making their use unnatural and non-user friendly. If users are not able to easily understand their structured input, they encounter difficulty in figuring out how to make a query using such a system.

An ideal interface should be able to infer what users want, based on the natural language used by the user(s), be it in the format of textual input or speech.

This project proposes and implements an intelligent chatbot system called 'SIA Assist', that can be primarily used to obtain information on all aspects of Singapore's Changi Airport.

Our proposed chatbot shows true effectiveness when the system has access to user's travel information so that his responses can be personalised - either by integrating the chatbot with an airline application or obtaining his travel information in some other form. In the demo, we have used dummy user data to showcase this personalization feature.

The purpose of the chatbot system is to allow users to search for common information about Singapore's Changi Airport in a more efficient and responsive way that accommodates for differences in syntax of questioning grammar but is still able to return a relevant response based on keywords in the query via Natural Language Processing (NLP). User(s) can also make use of suggestions given the chatbot. The effectiveness of such a system is reviewed based on efficiency of performance and effectiveness of functionality. The technology we utilised include node.js and Google Dialogflow.

Based on the review, we suggest some future proposals to improve the robustness of the system.

1. BUSINESS PROBLEM BACKGROUND

One of the more impactful and common applications of the chatbot technology is the information retrieval. Normally in searching for specific information on websites, the user must navigate and scroll through a large set of information to understand the same. This process can be time consuming and puts a cognitive load on the user, which in turn can affect the user's experience.

The ideal way to improve this difficulty is to leverage the affordances of a chatbot technology. Such chatbot systems can provide both voice and text assistance by a process of information extraction based on the user's input to return back to the user information that either directly responds to the user's query or at least showing a range of information relevant to the original query.

There are numerous avenues where this kind of technology can be implemented but none more so than enabling user's experience in a certain area which is always bound to be crowded.

Our proposed system implements the chatbot technology to help improve and personalize the user's experience at Singapore's Changi Airport.

It is seen that landing at the airport almost always affects a person's thought process because of the enormous environment. Due to this, the users are not aware of the attractions/facilities available. Generally, there are about an average of 5M people visiting Changi Airport every month - which ultimately does pose a large-scale problem with regard to the same. The current information accessible is not structured and thus puts passengers in a dilemma.

Our proposal would be valuable during passengers' time at Changi Airport.

2. OBJECTIVES AND SUCCESS MEASUREMENTS

2.1 OBJECTIVES

The objective of this project is to create an intelligent chatbot that will personalize the information to be given to each user and also answer any enquiries related to anything particular in the Changi Airport.

The primary target audience of our system are passengers who arrive or depart from the airport (including transit passengers).

'Sia Assist' aims to provide a solution to two challenges currently faced:

1. **Scalability** - where personalized interactions are challenging to support at scale due to the nuances of human patterns in communication which varies depending on individual and cultural differences
2. **Speed** - where in this current world, users expect instantaneous responses and services to their queries. Such solutions will free up expensive resources for more complex and high value-adding tasks.

Moreover, the chatbot will serve 24/7 to users, reducing operational and service expenses while increasing user engagement and touchpoints.

2.2 SUCCESS MEASUREMENTS

It is vital to understand the success of the system based on a few measures.

There are a few key measures of our chatbot:

1. Whether the chatbot was able to understand the user
2. Whether the chatbot was able to respond to the specific question being asked
3. Whether the chatbot was able to present the related information
4. Whether the chatbot was able to provide alternatives in case information related to user's query was not present.

3. SOLUTION

3.1 TARGET AUDIENCE

The major target audience would be the passengers arriving to or departing from the Changi Airport (including transit passengers) and require assistance to navigate the numerous facilities and attractions available.

.

3.2 DATA QUALITY AND KNOWLEDGE BASE

All information used to build 'Sia Assist' is from the Changi Airport's official website. It is assumed that the data present on this website are correct.

3.3 PROJECT SCOPE

In this project, a chatbot system based on the Changi Airport Website is performed using Google Dialogflow. 'Sia Assist' focuses on giving accurate responses related to various areas of the airport such as Baggage Assistance, Immigration, Restaurants, Attractions and Facilities aspect of the Changi Airport.

The intent-based system can answer question with context via rich messages response as well as voice format.

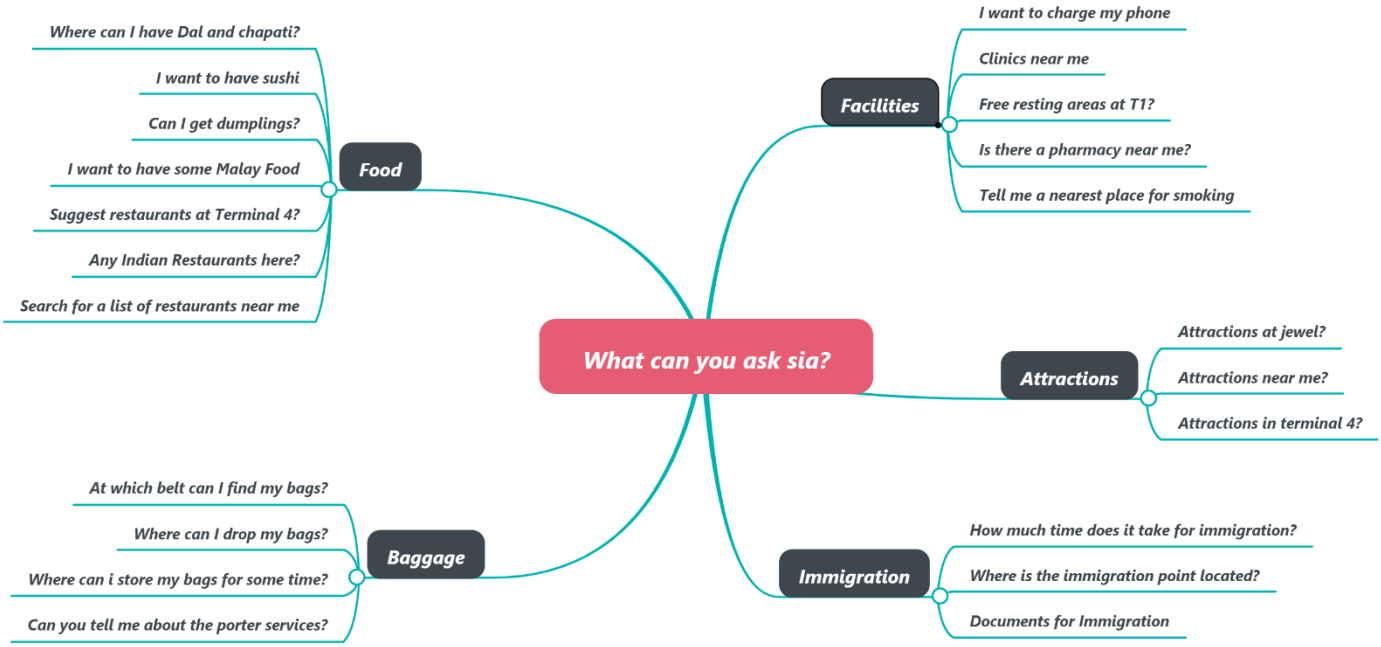

Fig 3.4 Domain Model of Sia Assist

The domain diagram above arranges the factors affecting user's question flow in a structural format. This chatbot can be broken down into multiple layers of different components before arriving at the answers. These question flows are gathered from users having a proposed question flow and represent their inherent preference.

4. IMPLEMENTATION

4.1 DIALOGFLOW SETUP

As Dialogflow provides the understanding of natural language it makes it easy to design and integrate the conversational user interfaces. SIA Assist has been built using the Dialogflow. The training of the chatbot has been completed using the intents which categorizes the end-user's intention for end user's intention. For the project, multiple intents have been created to well recognize user's questions and answer them by invoking the webhook call.

Dialogflow provides accessibility to features such as Intents, Entities, Fulfilment and Integrations which have been used in the project. For training the intents, such as Restaurants, Attraction among others, the training phrases has been provided to the intents, and external webhook call has been enabled in each of the intents to make use of backend call.

To provide the accessibility of a backend server, the URL has been provided in the URL section of Webhook in Fulfilment. As each intent parameter has the type called entity which dictates how exactly end user expressions would be extracted. Therefore, entities such as Baggage Entity, Cuisine Entity has been added to provide better results. As for each of the question asked to the chatbot, the Chatbot will match the entity of the intents based on the training in the Dialogflow, then the parameters will be passed through the webhook call to the backend of the project.

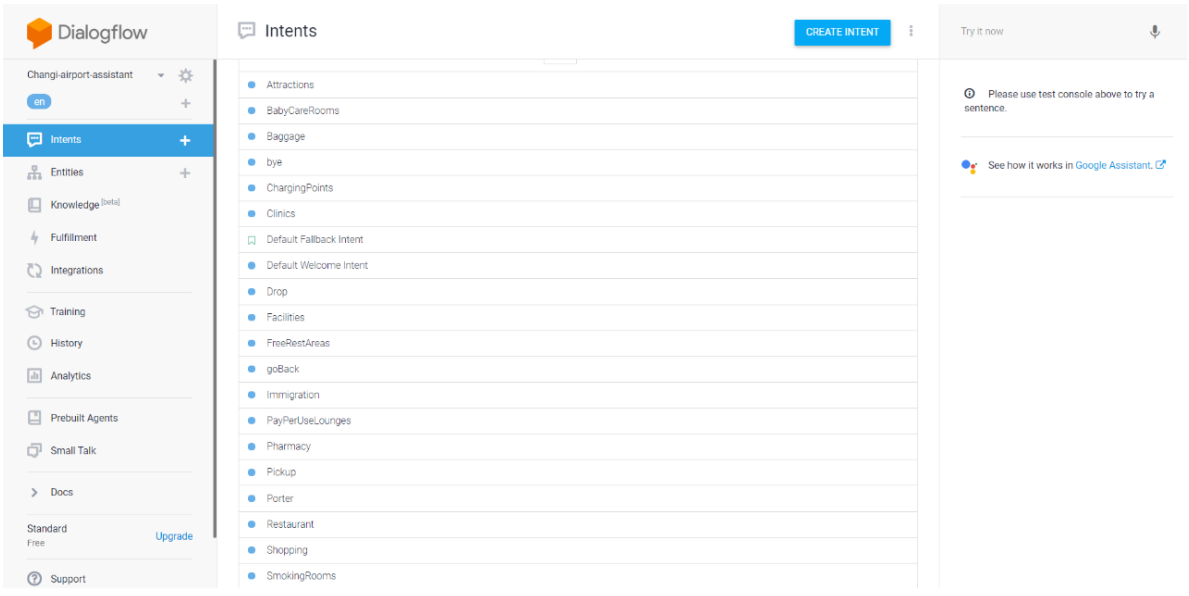

Fig 4.1 Dialogflow Home Page with Intents Displayed

4.2 GLITCH ENVIRONMENT

Glitch is a cloud environment that provides a server to host node.js applications. It provides inbuilt coding environment and auto-deploy features with no configuration needed. The backend code exposes a REST endpoint that enables Dialogflow to connect with it and get responses.

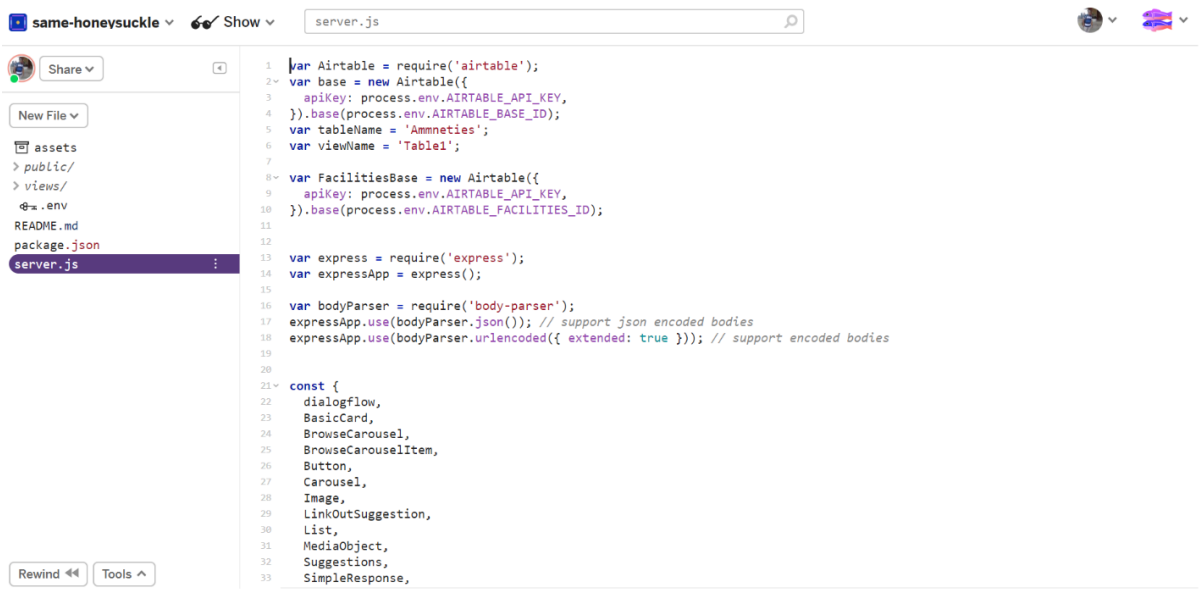

Fig 4.2 Glitch Coding Environment

4.3 AIRTABLE

Airtable provides an easy to use cloud environment with access to a database to control storing and retrieving huge amount of data without the typical hassles of a traditional relational database system. It also allows it to be used as a spreadsheet so that anyone, without technical knowledge of databases can update the tables. It provides the basic features in it's free tier.

It also allows accessing tables via REST APIs. Airtable provides a JavaScript library using which it an be accessed with ease from a node.js backend. The node.js server on Glitch is able to communicate with the database and make queries for the retrieval of necessary data requested by Google Dialogflow This enables us to provide real-time updates on the application that has been deployed using Airtable.

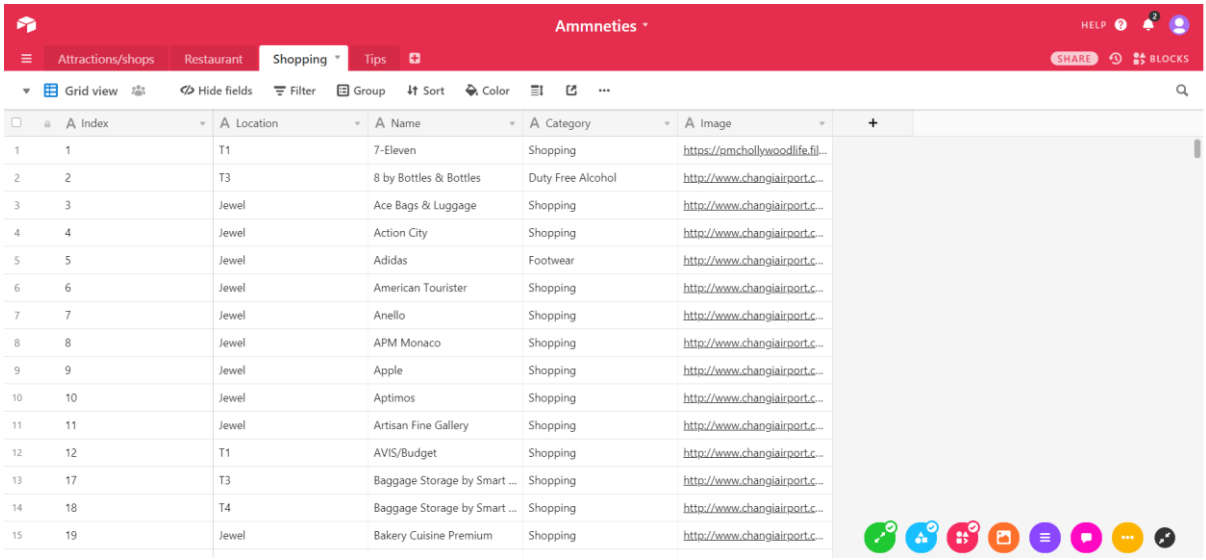

Fig 4.3 Airtable Database Table

4.4 UI TEST

To get a good user experience, we have used the Google Assistant platform provided in Dialogflow.

4.4.1 GOOGLE ASSISTANT

Google Assistant is the virtual assistant by Google, which provides the flexibility and ease to be integrated with Dialogflow. It provides the interface to use the Chatbot which can be deployed in very few steps. The Google Assistant is already accessible on most of the android smartphones and can be downloaded for iOS too.

It provides speech as well as text response along with Basic Cards, Chips, Button, Carousel and many other UI features.

The ease of use, flexibility, scaling are some of the features provided by Google assistant after the application is deployed.

In the below diagram (Google Assistant Test Environment), ‘Sia Assist’ successfully replied to the user’s greeting while also providing the various suggestion chips for the user to choose.

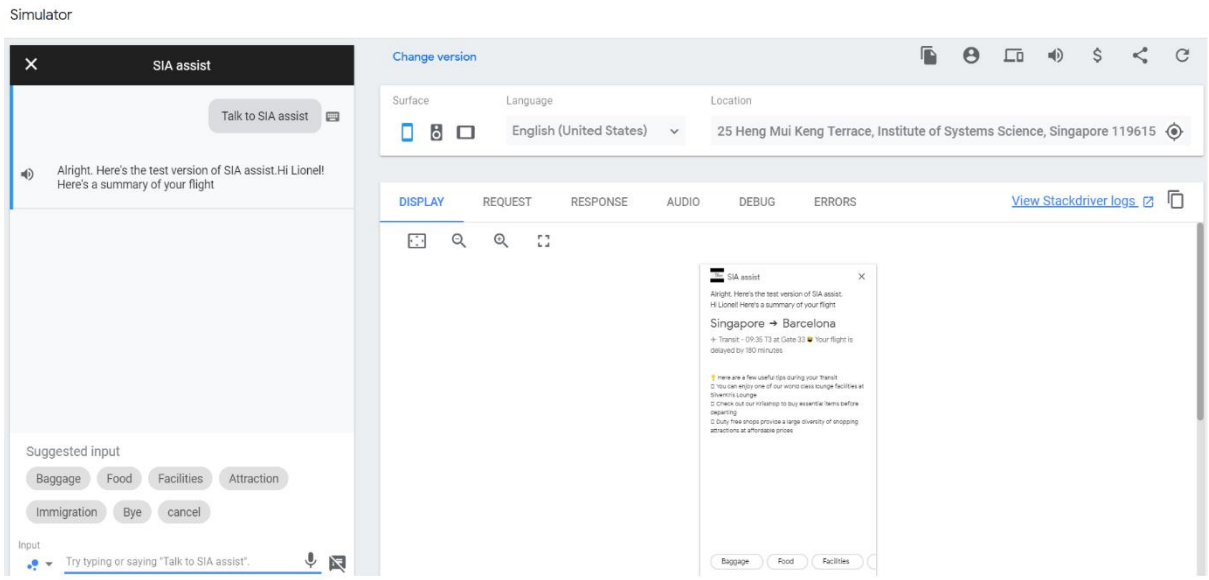

Fig 4.4 Testing the app on Google Assistant

4.5 SYSTEM ARCHITECTURE

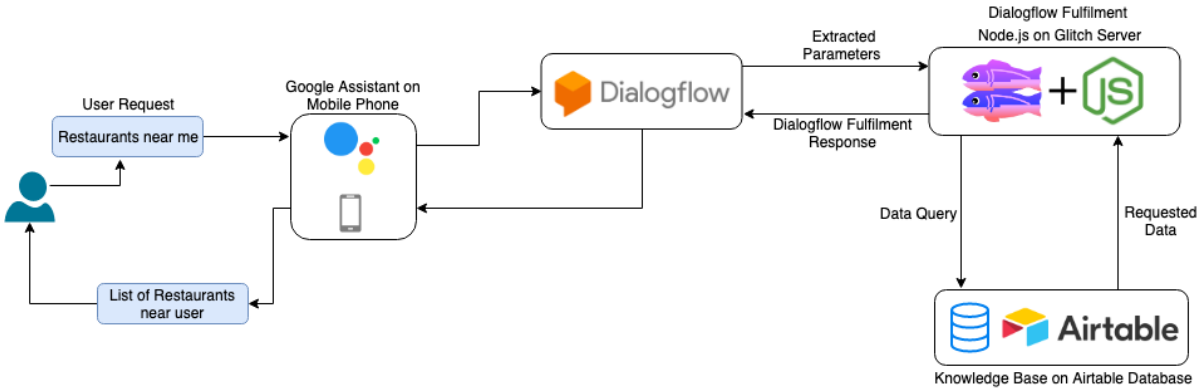

Fig 4.5 System Architecture Diagram

Figure 4.3 shows the system architecture diagram of 'Sia Assist'. It illustrates how the different components interact with each other through Dialogflow. After the user keys in the inputs on the Google Assistant using a mobile phone, Dialogflow will receive the input. Thereafter, Dialogflow will extract the intents and parameters from the sentence and feed it into the fulfilment unit, which in this case is the node.js server.

After the node server receives the request, the correct intent can be detected, and a query can be made to the Airtable database with the parameters provided by the user.

Once the data is received back, it can be cleaned and structured into a user-friendly response like a sentence or a carousel.

Eg. In this case, we have shown that that if a user request for ‘Restaurants near me’, then the query will go through the entire process, at the end of which Google Assistant will obtain the answer to return the List of Restaurants near the User.

5. PERFORMANCE AND VALIDATION

We perform validation on 2 different scenarios to ensure 'Sia Assist' provides the correct expected output.

Scenario 1

Scenario 1 is a passenger who is departing from the Changi Airport but is facing a delay on his flight. He now wants to explore the various options at the airport to use the amount of the time he has. He has various options between Restaurants, Attractions and Facilities to choose from. This could be done either through voice or textual responses to the chatbot. Below images, will show how the user could possibly interact with 'Sia Assist'

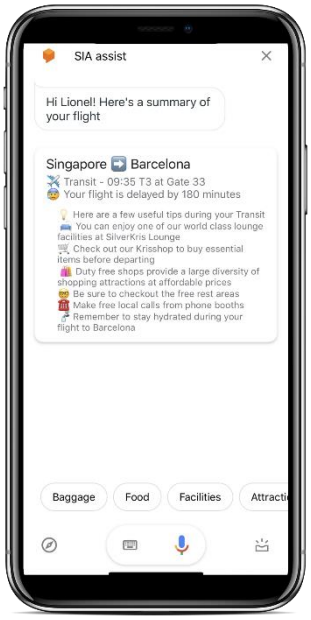

Fig 5 Data Retrieved from Passenger's Travel Card

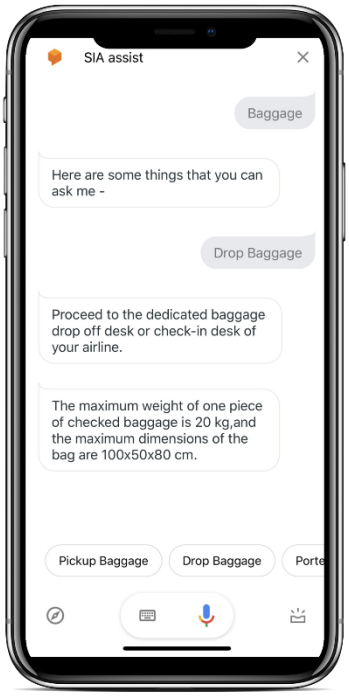

Fig 5 From Previous options, Passenger selects Baggage -> Drop Baggage

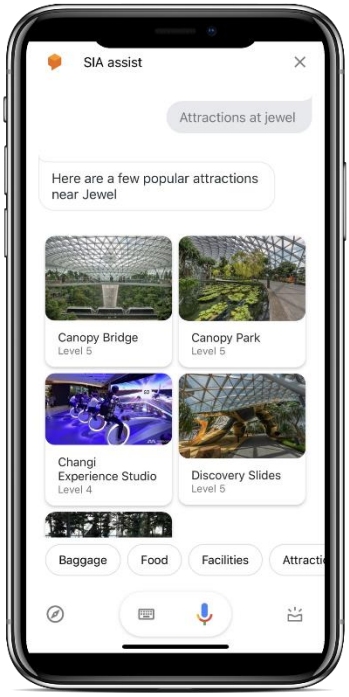

Fig 5 Post obtaining baggage information, passenger is provided details of attraction by the chatbot

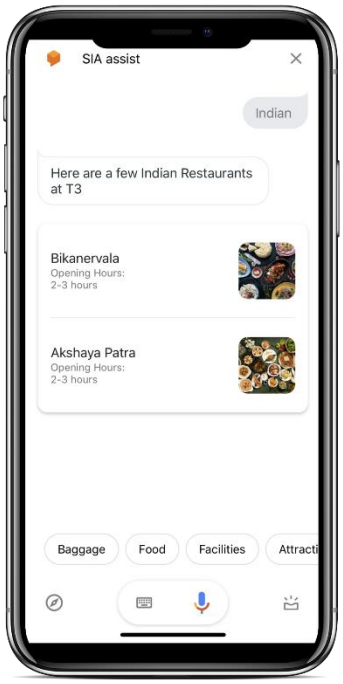

Fig 5 Furthermore, passenger is provided with specific restaurant information on requesting for the same

Scenario 2

Scenario 2 is a passenger who is arriving to the Changi Airport. He now wants to explore the various shopping options at the airport. Though he has various options, he possibly wants to look at specific items that he's interested in. Due to the difficulty in navigation, the chatbot can definitely provide assistance in this case. Below images, will show how the user could possibly interact with 'Sia Assist'

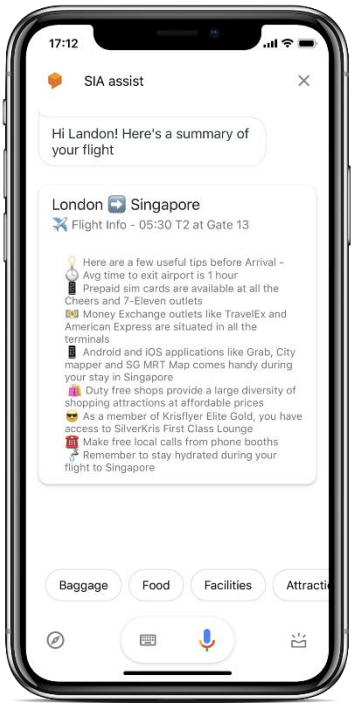

Fig 6 Incoming passengers' travel card is obtained and shown by 'Sia Assist'

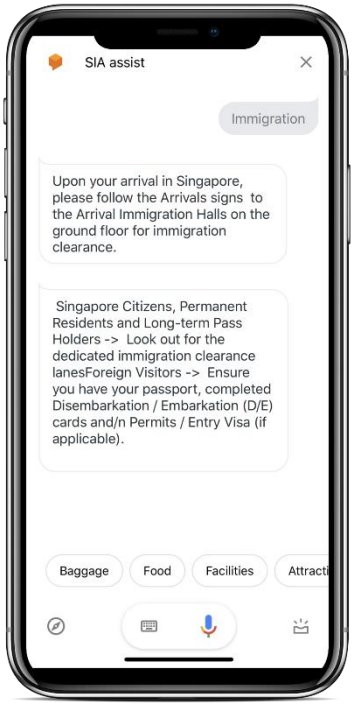

Fig 6 On passenger's request, the immigration details are brought up

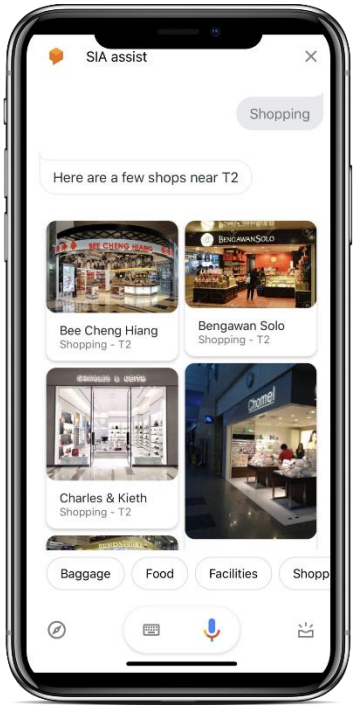

Fig 6 Passenger is shown the popular shopping outlets near his location on requesting

6. CONCLUSION AND FUTURE STEPS

In this project, a retrieval-based chatbot system is designed and developed via Dialogflow to aid passengers in navigating the Changi Airport and explore the massive options available at the airport.

Firstly, the chatbot is trained with more than 100 possible utterances that are annotated with intents and tagged with entities. Then, when a text input is received, DialogFlow will detect its respective intent and entity and subsequently, the fulfilment engine is enabled to retrieve and return the relevant information and response, respectively, through a http webhook. The information retrieval is performed via node.js. With such a system, the responses are standardised, less erroneous and are capable of performing with high accuracy. It works very well for such specific closed domain, and thus customer satisfaction and attention can be improved.

Therefore, this chatbot can serve as an effective customer service tool and the valuable human resources can be reallocated to handle more complex tasks.

The inherent limitation of the Information Extraction approach used to create the chatbot is it's lack of extensibility. Currently, the scripts used to extract information from the Changi Airport website are to be modified to extract data from similar websites of other airports.

Future enhancement of the project could involve, including data from more airports to integrate into the app for a seamless transition of information from one airport to the other.

7. REFERENCES

"Information Extraction from Changi Airport Web Pages"
https://www.changiairport.com/
https://www.jewelchangiairport.com/

"Knowledge Connectors | Dialogflow Documentation | Google Cloud"
https://cloud.google.com/dialogflow-enterprise/docs/knowledge-connectors

"Glitch | The server"
https://glitch.com/

"Airtable | The Database"
https://airtable.com/