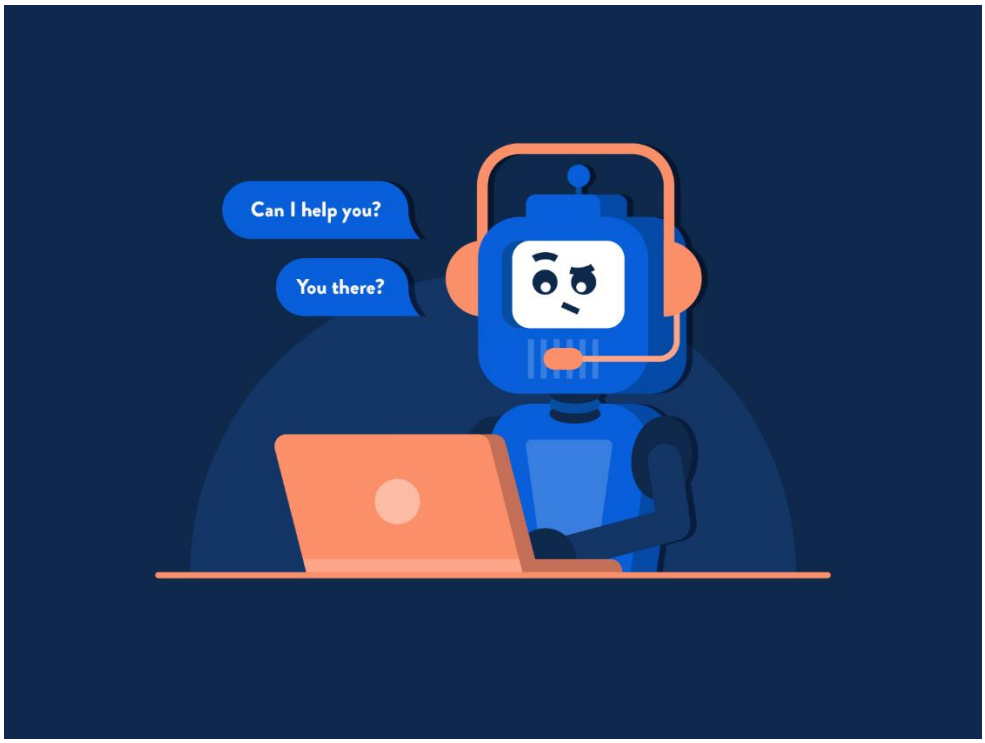
MASTER OF TECHNOLOGY
(INTELLIGENT SYSTEMS)

PROJECT REPORT (CA2)

SARA (Search And Retrieve App)
An Intelligent Chatbot for Searching Airbnb



TEAM MEMBERS

YONG QUAN ZHI, TOMMY (A0195353Y)
LIM LI WEI (A0087855L)
PREM S/O PIRAPALA CHANDRAN (A0195324A)

Table of Contents

1) Business Case

Chatbots which also known as “conversational agents” are basically software applications that mimic written or spoken human speech for the purposes of simulating a conversation or interaction with a real person. Currently there are two primary ways chatbots are offered to users which are web-based applications or standalone apps.

Intelligent chatbots are a growing trend for online consumer experience to motivate users to return to an online service due to its efficiency and efficacy in giving the consumer what they want. However there is also the need for the user to be able to craft the query in a reasonable semantic style understandable to the intelligence of the chatbot to make meaning out of it. Based on a survey of innovative applications of chatbots for online user experience[1], it is found their strength lies in making meaning out of human queries for automating repetitive tasks which may sometimes be in a batch format.

Natural Language processing is used to parse the data into an intelligible format that can be then acted upon based on interpreting the keywords as intents expressed by the user in the query to perform some actions. In Fig 1 is a representation of the flow of textual data from the front end of the chatot interface to the Natural Language Processing Layer containing the engine to make meaning from the query.

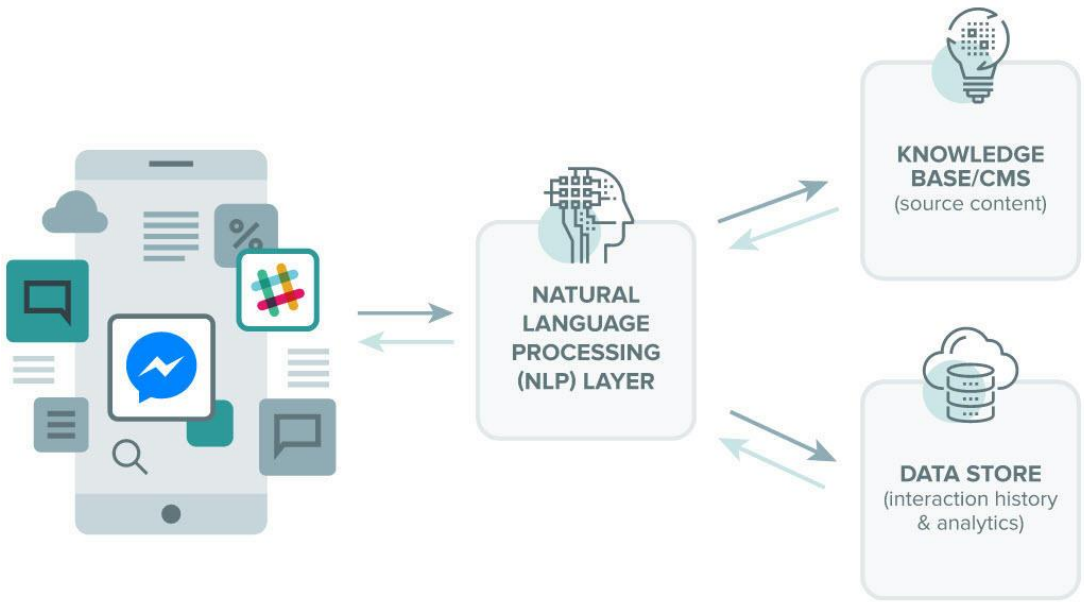


Fig 1: Overview of Chatbot Architecture for Chatbots using NLP
Retrieved from: https://bigdata-madesimple.com/how-do-chatbots-work-an-overview-of-the-architecture-of-a-chatbot/

How the NLP layer processes the series of words in the query is to parse or separate the words into tokens or pieces that are linguistically representative, with a different value in the application. Following that are the steps in the order of :

1. Sentiment Analysis: It will study and learn the user's experience, and transfer the inquiry to a human when necessary
2. Normalization: This program model processes the text to find out the typographical errors and common spelling mistakes that might alter the intended meaning of the user's request.
3. Named Entity Recognition: The program model of chatbot looks for different categories of words, similar to the name of the particular product, the user's address or name, whichever information is required.
4. Dependency Parsing: The Chatbot searches for the subjects, verbs, objects, common phrases and nouns in the user's text to discover related phrases to that which the users want to convey.

So chatbots can be regarded as a credible alternative to a human respondent with enough intelligence to answer like a human to any questions it can make meaning from. It is noticed that often in making any queries online, a user is often caught in a situation where they are conversing online and having to switch between apps in searching from a website using information from an online chat. A simple scenario to illustrate this would be the following

---

Scenario

A user is in the middle of a texting chat about travel plans with a friend. He is interested in getting some recommendations about a possible destination. He has to leave his chat app and then navigate to the Airbnb website to make a search matching his criteria (Airbnb is an online marketplace which lets people rent out their properties or spare rooms to guests. There's plenty of criteria to list for/search a property)

While searching on the Airbnb site, his friend at the messenging app has suggested they bring their family members along, which the user only comes to find out when

he returns to the chat. This now means the user has to conduct another search based on new criteria.

As can be seen, the above scenario is a common experience for many mobile users looking to make online bookings while chatting on their mobile device. In fact based on a quick verbal qualitative survey we did with our classmates, the feedback we got was the following

- frustrating and irksome to keep shifting focus between the chat app and the search online.
- the content that the search returns is not easily shared to other apps

For our business proposition for this project, our team sought to see how we can make this a more seamless experience that allows the user to switch from chat to search in a more intuitive way without losing conversational flow in their chat and being able to share their search findings with their friends within a chat.Using the technology we have were familiarised with in Intelligent Automation Automation (IPA)

Some of the more important aspects of the user experience would be the ability to share relevant information immediately within the chat such as a list of possible links, map locations and photographs of the destinations so that a quick and informed choice can be made among the list of suggested destinations that is returned to a user using this product.

A test of the mobile user experience of the *Airbnb* website using an online mobile device emulator shown here(click link) shows the limitations of the responsive design of the website1 as also shown in the screenshots below showing the transition interface screens in navigation during a simple search.
https://responsivewebdesign.com/podcast/airbnb/

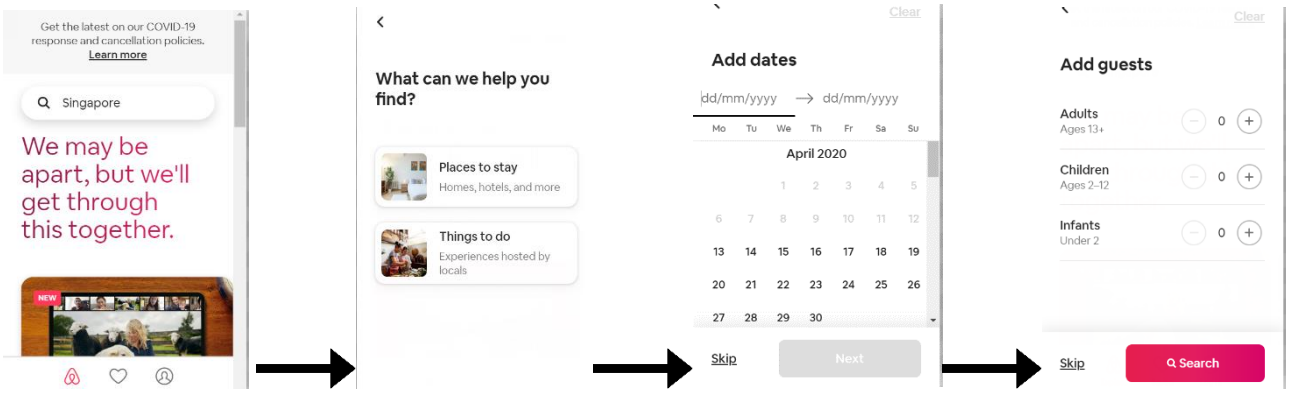


Fig 2 : Screen interface flow of a search done on *Airbnb* website

As can be seen from the flow in the screen interface, a user will have to navigate through a set of four screens requiring keying in a criteria for the search at each stage to finally get back a return of results. There is also currently no online chatbot on the website for users to make any query.

This is a time consuming process that will pose a challenge to a user when trying to balance this while texting. To address this, we posed ourselves the design brief for the mobile experience by asking two fundamental questions.

1. ***How can this search be automated based on a set of input criteria within one screen ?***

2. ***How can the returned results be shared more intuitively using a mobile device ?***

To develop a better understanding of how to develop the product, a Use Case diagram was created to illustrate the different features necessary for the design of the user experience as shown in Fig 3 below.

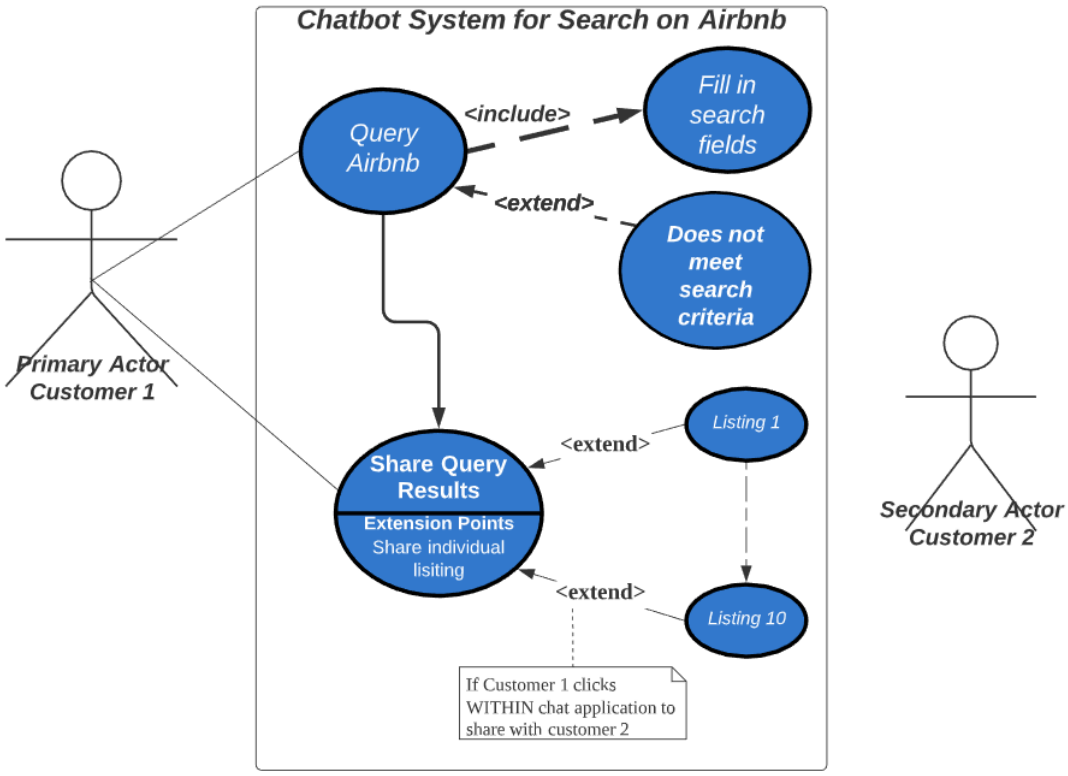


Fig 2 : UML (Unified Modelling Language) Case Diagram for Chatbot system

Link to view diagram: https://app.lucidchart.com/invitations/accept/30be15f9-7165-4726-ae23-7d1a2b0d9286

The use case diagram allowed us to understand the different ways in which the user will interact with the system and consider what features to include in the system design to address those interactions.

2) Product Plan

For the success of any project to be completed on time and within budget, it would be prudent to have a project plan for our product completion on time. For a minimum viable product to be launched, we referred to the readings we had covered for the course and used the approach as advised in "How to set up a minimum viable product" to answer these fundamental questions

1. **Start by framing the problem**

   To have better clarity for earlier problem articulation, we framed our problem as being how to create a user experience where search can be automated based on a set of input criteria within one screen? This would be identifying what front end technological platforms we could use in our chatbot as a messaging interface.

2. **Identify your users**

   As for our users, they were basically mobile phone users who are in the habit of searching for travel accommodation on Airbnb website while balancing the act with texting or messaging

3. **Understand your users**

   In understanding our users, we realised we are trying to make the interactions of searching on Airbnb to be seamless with texting so that possibly the search can be conducted in the background as the priority for our users is to not lose the flow of texting

4. **Ideate on potential solutions**

   To come up with possible solutions, we had to think of how to integrate the various technology together in having a front end messaging service that can

integrate with conducting a search on Airbnb. We had to consider how the two distinct platforms could talk to each other and how the automation could be achieved by using RPA tools and which RPA platform would be optimal so not to return errors in time outs or

5. **Define your MVP — the Minimum Viable Product**

A full product would be able to search different travel accommodation websites, but for our minimum viable product, we wanted to focus on showing how the chatbot could work for one as a proof of the concept and to use that an example to study how the user experience could be enhanced.

We outlined how we could approach designing it by researching and comparing industrial standard existing project management methodology models of AGILE vs SCRUM vs Waterfall vs KANBAN[2].to determine which of these would be a suitable model to adopt for our project.

Based on the attributes of this project being

- more iterative in nature in continuous revisions in improving the user experience
- short time frame for implementation (slightly less than a month)
- small team size (3 members of which 2 focused on development and 1 on documentation & marketing)

we decided our product plan approach had to be more flexible.
It cannot be said that we followed strictly any of the above mentioned models. Our team approach was more of a hybrid blend merging aspects of each methodology which reflected more of features of an Agile approach as we were uncertain if the system design would be effective being the 1st time we are doing it. However familiarity with the different components in the system design gave us confidence that we would be able to achieve it.

3) Market Research

For any product to succeed, it would be prudent to first commit to market research to find out what are similar products already out there so as to carve out a niche for our own product that creates enough differentiation in terms of the service that is offered so as to solidy a customer base from which the product can spread by word of mouth to others.

A survey of existing technology on the online market using Google Search shows that there are several chatbot designs for *Airbnb* created by users as listed and explained here

No	Link	Remarks
1	https://medium.com/@natehindman/airbnb-bot-e5da9f70a477	#Slack bot to conduct a direct search only for location through Slack
2	https://www.burnerapp.com/blog/2017/2/21/hostbot-a-burner-line-for-vacation-rental-hosts	smart agent that automatically interprets and answers the “frequently asked questions” a

However all bots lacked the functionality and flexibility we aimed for in our design, which was to allow all the query to be entered in a NLP(natural language processing) format recognisable to the chatbot satisfying all the required search filters to return a set of results. and most importantly integrable within a messaging service used for texting.

4) System Design

In this section, a brief description is given of each aspect of the different components of the overall system design Basically, the role of that component is explained and the technology in terms of choice of the platform used to implement that respective component.

Front End
(Choice of Platform: Telegram)

Telegram is a popular messaging app that has the ability to allow developers to create chat bots within seconds and allow consumers of the app to add and start using the bots. They also provide APIs and have python libraries that facilitate communications between the chat bot and a developer's backend server. As such, it was the perfect platform to accept user inputs, process the inputs and display responses to them in a user-friendly way.

There are a couple of competing platforms which have been considered over Telegram such as Whatsapp, Facebook Messenger and WeChat. However, Telegram as ultimately selected as Whatsapp has legal policies against Bots usage, and Wechat is not as accessible. Facebook messenger has a few nifty features which may be able to display the images better using carousel, however, after doing a market study, more people use Telegram compared to Facebook Messenger, which is why it was not chosen as the platform for this project.

NLP (Natural Language Processing) Agent
Choice of Platform: DialogFlow (https://dialogflow.com/)
DialogFlow is a development suite for creating conversational interfaces for websites, mobile applications, popular messaging platforms, and IoT devices. For this system, it is used to
create the intelligence for the conversational chatbot to enable the interaction between the user and
It can be used to build interfaces (such as chatbots and conversational IVR) that enable natural and rich interactions between your users and your business.

Dialogflow Enterprise Edition users have access to Google Cloud Support and a service level agreement (SLA) for production deployments.

By using Dialogflow, we are actually exploiting Google's huge databases of existing entities for items such as Countries/Cities and DateTime. This relieves us of crafting our own entities manually, making it easier to define the rules to trigger the intent. Dialogflow is also user-friendly and easy to use over a web interface, and recently it has grown to support 3rd party integrations, such as telegram. This allows us to seamlessly meld the 2 platforms together to develop a powerful solution. One of the drawbacks of using Dialogflow is that Google will tend to change business models frequently. As such, Dialogflow might end up being a paid software in the future.

Web Framework to host website
Choice of Platform: Flask

Flask is a web framework used in our project as a backend server. It is the centerpiece of our system which orchestrates efforts between each technical component of our solution. It is firstly used to accept input data derived by Dialogflow from user texts. It will pass these inputs onto the RPA functions which will then open the AirBNB website and start scraping for data. Once this data has been gathered, it is further processed with certain functions defined within flask to make it more structured and meaningful. For instance it will organize data in a summarized and meaning format for easy readability. It will also use each airbnb listing to find meaningful venues around it via third-party API calls. Finally, it will send the formatted data back to the telegram bot and it will display these outputs to the user.

So in this case, it is configured as a middleware for communication between Dialogflow and the Python Script, to pass variables around across 2 different environments. The initial plan is to host the Flask script on Heroku. However, TagUI does not support headless chrome execution on Python. Therefore, the script has

to be hosted on a computer that has the ability to open chrome browser (Can be a VM, but cloud VMs are not free).

RPA Software

Choice of Platform: TagUI

An Open source tool created by AI Singapore, TagUI is a simple RPA tool for making software robots, especially for doing repetitive, structured, and rule based automation over a website. It is lightweight, supported for python usage, and allows effortless crawling of data, as well as data entry into any website. In this case, we are applying its functionality on the AirBnB Platform.

However, as a lightweight tool, it is not able to accomplish functions that industrial RPA softwares can achieve. One of its prominent competitors in this project will be UIPath, which supports smart recording, comprehensive selector manipulation and webelement interaction, driven by a flow chart. Ultimately, TagUI is chosen for its lightweight framework that does not need much instructions to setup, and ability to perform web automation using XPath syntax to achieve all that is needed in this project. What's more, it is free, being open-sourced.

One of the pain points of using RPA stems from the dynamic nature of AirBnB's website's User Interface. Whenever a huge change is made to the website by AirBnB, we do need to relook into the RPA script and change it. This actually increases the maintenance effort to sustain the bot.

The overall system design is represented by the diagram below in Fig. 4

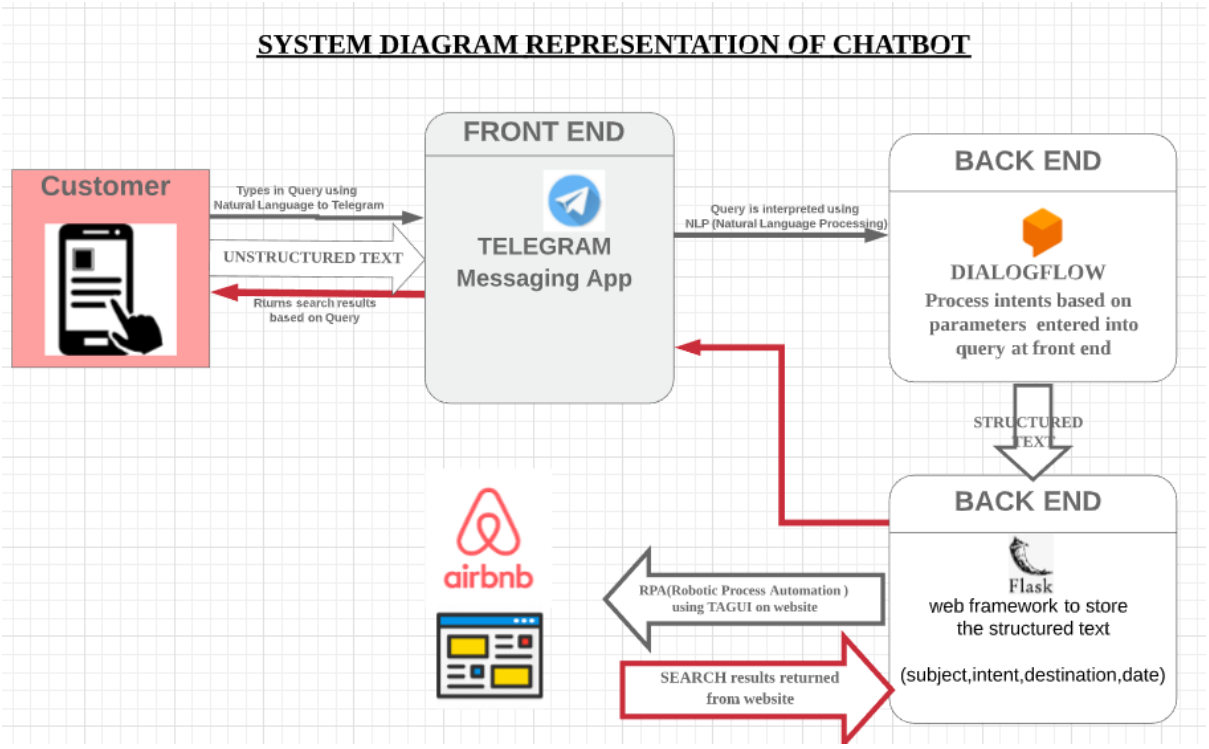


Fig 4 : System block diagram for data flow within Chatbot system

Link to view diagram: https://app.lucidchart.com/invitations/accept/1b03dc8b-c9c7-4fb9-83f3-24e3c10dd3c4

Appendix 1: Installation and User Guide

Setting up the project

Please ensure you have the following ready before proceeding with the instructions.

- You are using python 3.6 or higher
- Telegram app has been installed on your android or iOS smartphone.
- You have a registered account on https://dialogflow.com/

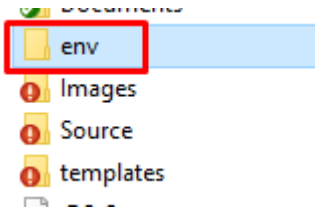
Note: During the course of this project, AirBNB changed their UI template a few times. Please note that the RPA component may not work if AirBNB has changed their template again. Please let the team know if this case occurs.

Once you have installed python, open a command prompt and CD to the project root folder("<filepath>/ ISA-IPA-2020-03-21-IS01PT-GRP-Etika-SARA_AirbnbAgent /SystemCode").

```
D:\>cd "ISA-IPA-2020-03-21-IS01PT-GRP-Etika-AirbnbAgent"
```

1. Create a python env. Once created, you should see an 'env' folder in your project's folder

   Use "python3 -m venv env" OR use "python -m venv env"



2. Activate the python env. Once activated, you should see (env) next to your command line.

   For Mac: "source env/bin/activate"

   For Windows: "env\Scripts\activate"

```
(env) D:\ISA-IPA-2020-03-21-IS01PT-GRP-Etika-AirbnbAgent>
```

3. In the command prompt/terminal CD to the project folder "<your-file-path>/ISA-IPA-2020-03-21-IS01PT-GRP-Etika-SARA_AirbnbAgent/SystemCode". Enter "pip install -r requirements.txt" OR "pip3 install -r requirements.txt". This will install all the required dependencies.

4. Once installation is complete, type in " flask run" and press enter. This will deploy your server locally on your pc.

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

5. You need to open a channel to your computer so that dialogflow and the telegram bot can communicate with your server. There is an "ngrok_win" and an "ngrok_mac" file. Rename the file with your OS in use to just "ngrok"(*figure 5.1*). E.g If you are using Windows, delete the "_win" from "ngrok_win". Then open a new command prompt and CD to "<your-file-path>/ISA-IPA-2020-03-21-IS01PT-GRP-Etika-SARA_AirbnbAgent/SystemCode", enter "ngrok http localhost:5000". Take note of the https link, it should look something like this https://f34bb6f6.ngrok.io (*figure 5.2*).

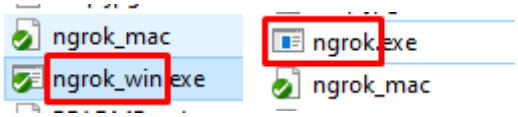


Figure 5.1 Renaming of ngrok file

```
Forwarding                    http://ca0ef6c2.ngrok.io -> http://localhost:5000
Forwarding                    https://ca0ef6c2.ngrok.io -> http://localhost:5000
```

Figure 5.2 HTTPS link

6. Go to https://dialogflow.cloud.google.com/ and log in with your account. Proceed to import the dialogflow agent(IPA-AIRBNB.zip) on dialogflow. To do this, first create a new agent and give it a name e.g. "AIRBNB AGENT"(*figure 6.1*). Once created click on the settings gear icon next the agent name and

click the “Export & Import” tab(*figure 6.2*). Click “Import From Zip” and select the IPA-AGENT zip file(*figure 6.3*). Type “IMPORT” into the text box and click “IMPORT” (*figure 6.4*).

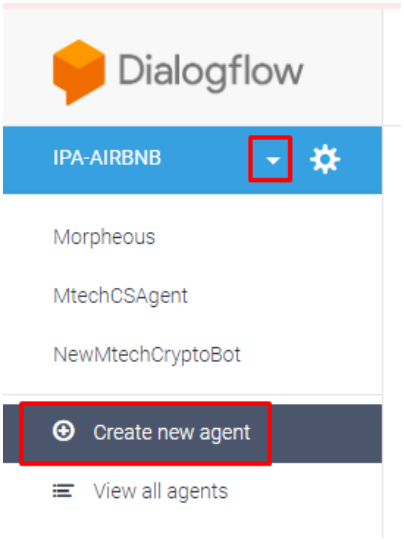


Figure 6.1 Create new agent

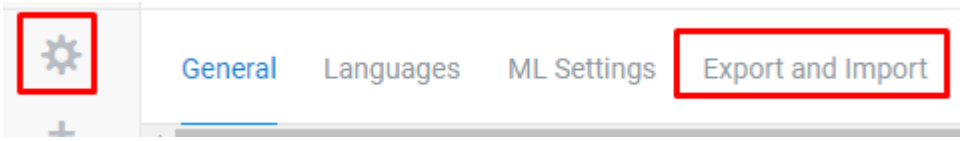


Figure 6.2 Import options

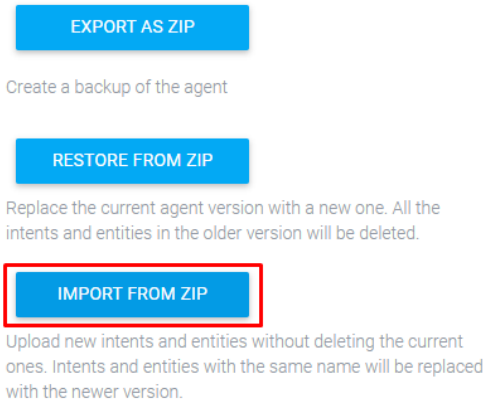


Figure 6.3 Import from zip

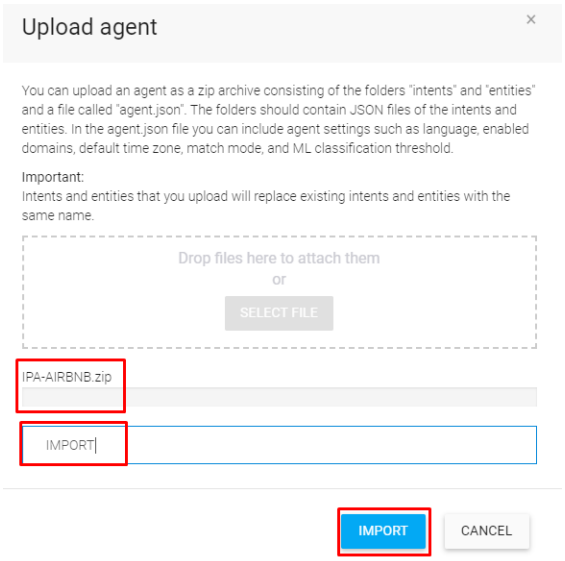


Figure 6.4 Upload options

7. Once the agent has been imported and training is done, click on the "Fulfillment" option on the left menu bar(*figure 7.1*). Enable the webhook(If it is not enabled), and copy and paste the ngrok https link on the URL field. Add in "/get_recommendations" right at the end of the ngrok link(*figure 7.2*). Scroll to the bottom and click save. Give it some time to save your settings.

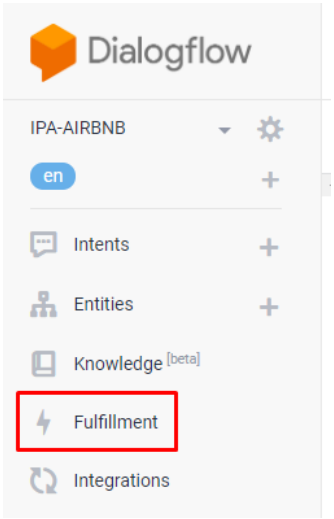


Figure 7.1 Fulfillment option

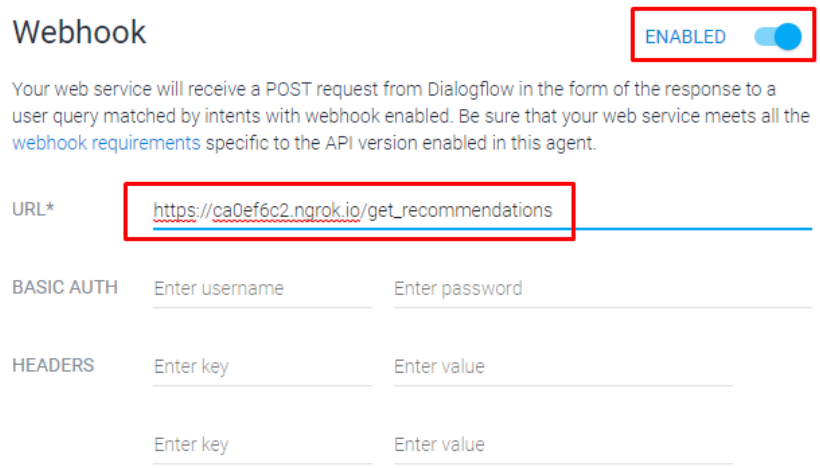


Figure 7.2 Enable and setting webhook

8. Set up the link between telegram and dialogflow. From dialogflow, click Integrations(*figure 8.1*). Then, check the Telegram box(*figure 8.2*). A pop-up will appear, enter the bot_token key and click start(*figure 8.3*). This will integrate the bot and dialogflow together.

Bot Token key: 1080638501:AAHhksiyyTfavxdcr8SXPZWFQpmV2Hx2mK0

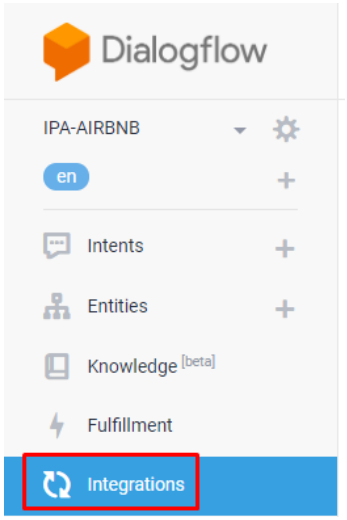


Figure 8.1 Integrations option

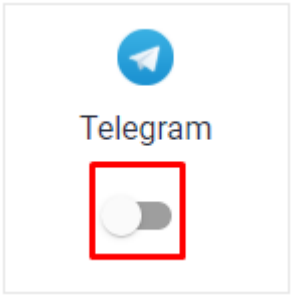


Figure 8.2 Telegram option

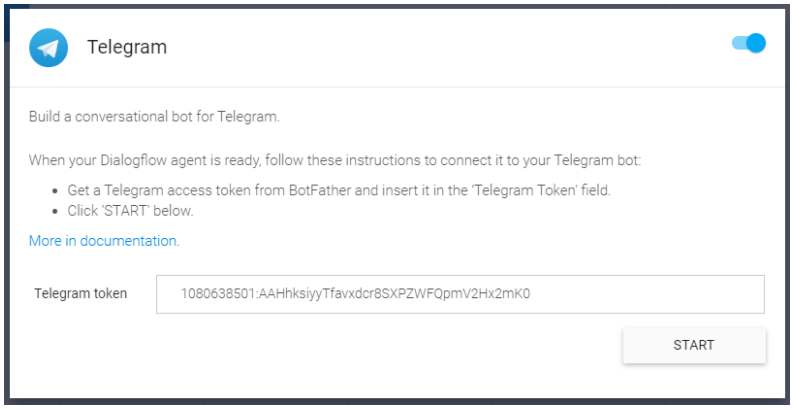
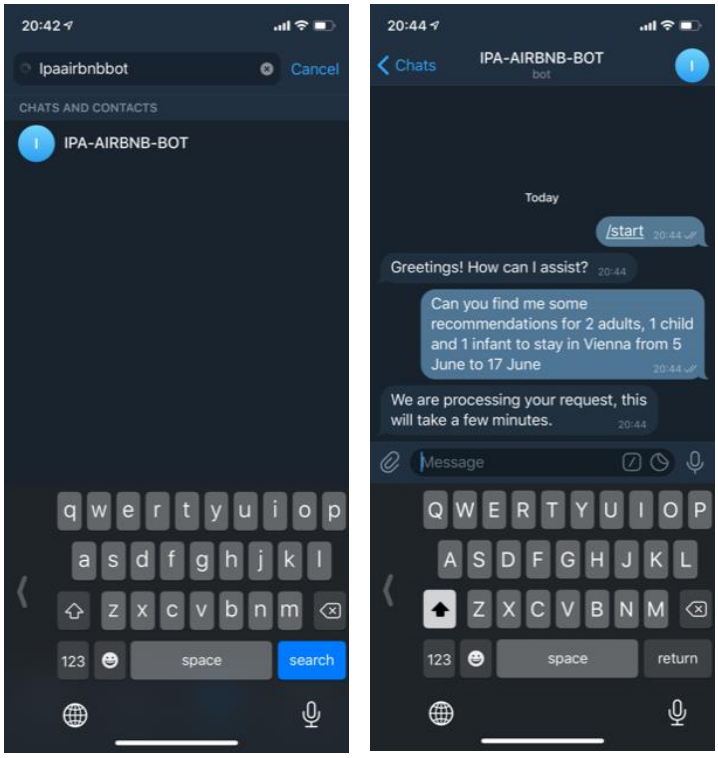
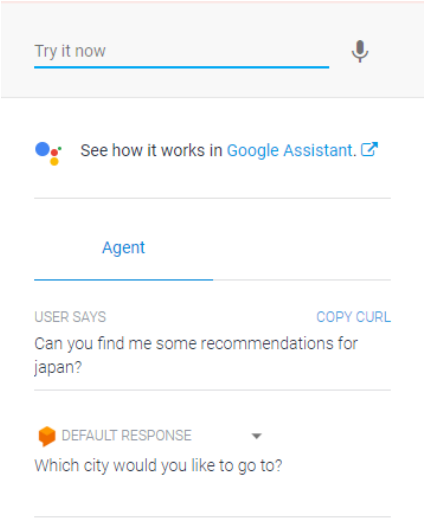


Figure 8.3 Telegram option

9. Now open up your telegram app from the smartphone. From the search bar of your chat page, type "IPAAIRBNBBot" click on “IPA-AIRBNB-BOT”. A chat window will be created with the bot, click 'Start'. You can now test sending some search queries such as "Can you find some recommendations for 2 adults, 1 child and 1 infant to stay in Vienna from 5th June to 17th June?". Alternatively, you can open the conversation with “Can you find me some recommendations” and the chatbot will proceed to ask you necessary questions before searching on Airbnb for your recommendations.



You can also test dialogs from the google assistant via the dialogflow website located on the right side of the screen.



Appendix 2: Individual Reports

Name: Prem s/o Pirapala Chandran

For this project, I undertook the following roles of report writing and creating the promotion videos for our app. These were done with feedback from my teammates, Tommy and Li Wei.

As a team we talked to different people to find their habits with mobile devices and came to an understanding of how a lot of decision making in online reservations through booking on websites can be a frustrating process when multi-tasking while chatting on mobile devices. There is a need to compare pricings, locations and accommodations and sharing such information with relevant parties whom we are texting to come to a mutual decision. As such a ideal approach would be to blend the two activities in a seamlessly, which is how we came to the idea of merging a chatbot on a cloud based messaging chat service app Telegram with an online lodging service Airbnb. I suggested the name “SARA” as a possible name for the product to my teammates which they were agreeable to as their efforts in designing the interface for the front end and channeling the data from the back end mimicked a SEARCH AND RETRIEVE APP function and the mnemonic was used to name our chatbot as such.

For representing the use case I created a UML(Unified Modelling Language) Use Case Diagram using Lucid chart to serve as an info graphic that depicted the scenarios of use of our chatbot

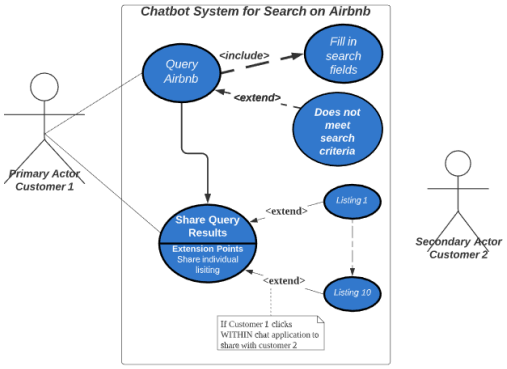


Figure 1:Use Case Diagram for chatbot design
https://www.lucidchart.com/invitations/accept/30be15f9-7165-4726-ae23-7d1a2b0d9286

To represent the system design, based on my understanding and feedback from my teammates about how they integrated the different technologu together, I used Lucid chart to create the following info graphic to depict the overall system design.

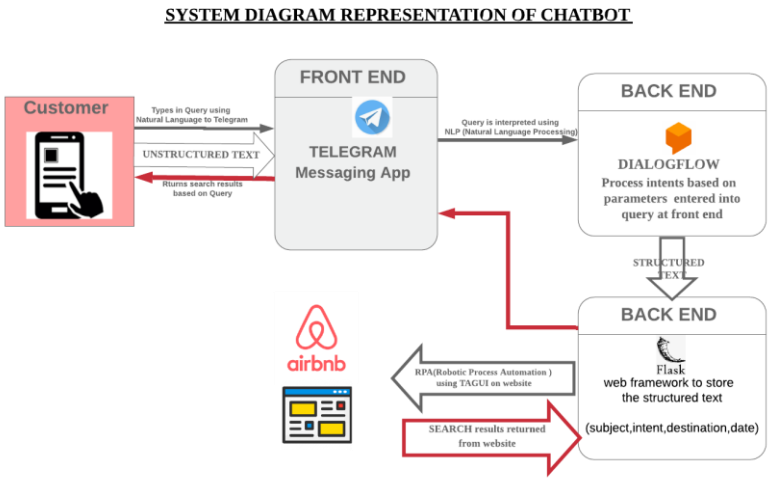


Figure 2 : Infographic for System integration in Chatbot Design
https://www.lucidchart.com/invitations/accept/1b03dc8b-c9c7-4fb9-83f3-24e3c10dd3c4

For the promotion videos, I scripted and created two animation based on my understanding of the design of the chatbot to convey the use qualities of the final design so that potential users could understand the benefits of the product and how to use it .The marketing animation video and the promotion video can be viewed here by clicking on the respective images

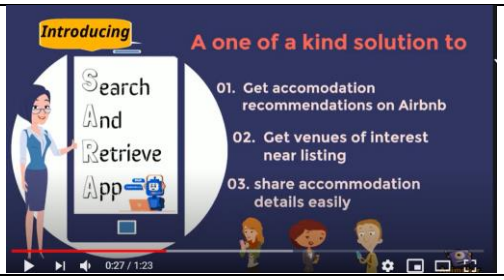


Figure 3 Promotion Video for Chatbot SARA

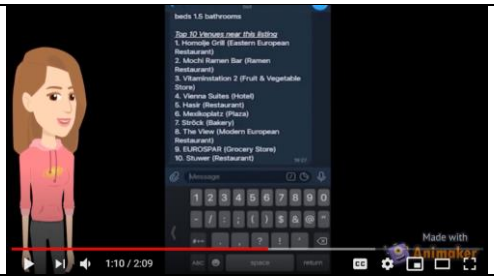


Figure 4 Chatbot SARA - How to use

All these skills I learnt can help me with app design and how to market apps in an appealing way to customers. In report writing , it is to craft a coherent report that can inform readers about the thinking and methodology that went into designing an app by understanding the technological explanation from the coding team and then rephrasing it for the layperson to understand.

Name: Yong Quan Zhi, Tommy

Personal Contribution to group project

1. Setting up of telegram chatbot
2. Setting up flask and creating functions within flask(index.py)
3. Setting up dialogflow to process user inputs
4. Creation of user guide

What you have learnt

For this project I have learnt about 2 types of AI tools/techniques. The first is developing RPA functions using TagUI to automate certain tasks that a user would normally carry out. As we mostly practice with websites, we learned about XPath and various programming techniques to handle exceptions, complete tasks and validate data. All of which are necessary skills to employ with real world projects. The second tool I learned was Dialogflow. Even though it was a skill acquired in the previous semester, this gave me a chance to polish my skills and try it out on a different use case with real world applications and to integrate it with other systems.

Besides these techniques, I have also picked up how to set up a flask backend server which has also given me a chance to polish my python programming skills. I had the opportunity to set up the necessary integrations between flask to dialogflow, FourSquare's API, telegram's APIs and my team member's RPA functions. All of which are practical skills needed for many software engineer roles.

How you can apply the knowledge and skills in other situations

As the main skills I've picked up focuses on RPA and dialogflow, I feel each of these can be applied easily for different types of job functions. For instance, at my company, there is a strong need to generate reports on a daily, weekly and monthly basis and have those reports transferred to a shared drive so that the respective

employees can review them. The RPA skills I have picked up are highly practical for this scenario and can save them a great deal of time from having to manually generate such reports and drag into a shared drive folder.

With dialogflow, I can see that this has good potential for SMEs to pick this up to create their own simple chatbot to provide helpdesk services to their customers. For instance, a local pc/laptop company can use dialogflow to integrate with chatbots from telegram/facebook messager/slack and allow users to get price checks, get PC/laptop customization recommendations, set up appointment for servicing or even to ask other frequently asked questions. Such a service can be available to everyone at any time of day.

Name: Lim Li Wei

For this project, as I am the most familiar with RPA related software in the group, I took up the role of implementing the RPA part, which will go into AirBnB to enter detail and return the values. In this project, I have utilized TagUI for its seamless integration into python scripts, taking in account that we will be using flask to host the script needed to communicate with dialogflow. I started with designing the framework for the main RPA script, so that the web designer (Tommy) can easily integrate my script as a library and call them as simple functions for use in the overall project.

Using an array of xpath selector operations, I managed to pinpoint all the elements and interact with them, respectively. This started from entering the city name, the dates and the number of people, to exploring the first 5 homestays and extracting their URL, pictures, description, cost and reviews, and even the map coordinates in longtitude and latitude, which will later come useful in the output part.

TagUI is not without its limitations. It is not able to identify and utilize complex selectors as easily as UIPath can, hence I had to compensate by putting to use the comprehensive knowledge on xpath selector taught in class to ensure that the elements that were used are unique, and are resilient to changes. However much I did, there is still a chance that AirBnB changes user interface on whim in production, causing the RPA script to fail. AirBnB also introduces a lot of Anti Robot elements, such as captcha for login (which I end up not doing), and at times, has more than 1 combination of possible xpath selectors for each element, in which I have to catch and account for all in order for the robot to work correctly. Even as of now, if Airbnb does another overhaul, our RPA script will fail. This highlights one of the disadvantages of RPA.

Comparing this project with my day to day work as a RPA consultant, I can feel the similar pain points during the making of this robot, as compared to using Kofax/Blueprism. As such, I learnt that traditional RPA tools are fundamentally the same.

Using the knowledge applied in this project, this will help me in my line of work. In RPA, one of the key factors is to choose the best selectors to use for each element,

to ensure the robot does not fail given some dynamic condition and stays resilient in production run.