# MASTER OF TECHNOLOGY USER GUIDE

Mini Museum Chatbot System


**TEAMMEMBERS**

XU JIACHEN LI XINLIN

MASTER OF TECHNOLOGY

# Contents

**1.0 System Overview** ..... 1
**2.0 Requirements** ..... 1
**3.0 Deployment** ..... 2
**4.0 Walkthrough** ..... 7
**5.0 Business Scenarios – Sample Input and System Output** ..... 8
**6.0 Appendixes** ..... 15

# 1.0 System Overview

Mini Museum Chatbot system is our team project of the Cognitive System Course, which is a chatbot system based on the Action on Google(Google Assistant), Google's DialogFlow and Python to provide the fundamental information and recommendation information of the National Museum of Singapore to its tourists. Our goal is to create an intelligent National Museum of Singapore chatbot System that will be better serve the tourists with Google Assistant and offer Information and services for visitors who come to visit the National Museum of Singapore through natural language or who want to visit the National Museum of Singapore.

# 2.0 Requirements

The project uses the Google Assistant as the frontend interface, DialogFlow as the NLU Engine and middleware and the Flask as the backend and the fulfilment of the DialogFlow.

Data Structure

- Requests and responses between the Google Assistant and DialogFlow follow the Action on Google Conversation Protocol
- Requests and responses between the DialoFlow follow the DialogFlow Conversation Protocol.
- Flask backend server provides the JSON format response follow the DialogFlow Conversation Protocol and Action on Google Visual Components Documentation

## 2.1 Recommended Test Devices

Mini Museum chatbot demonstration supports the following test devices:

- Android Phone with GApps and has installed the Google Assistant(Recommended)
- IOS phone has installed the Google Assistant(Recommended)
- Action on Google Simulator using the Google Chrome Brower (Can't get the full experience)

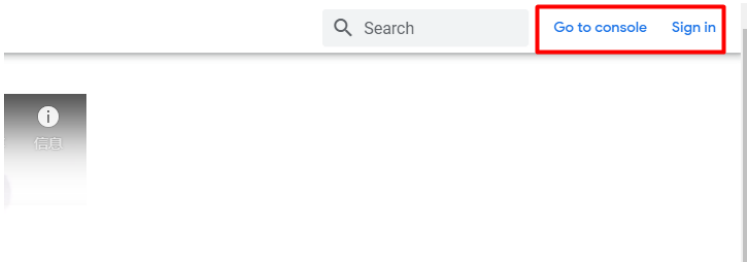
## 2.2 Dependencies

- Frontend
  - Google Assistant
  - Action on Google
- DialogFlow
- Backend
  - Python 3.7
  - Python Library
    - Flask
    - Flask-Assistant

        - Panadas
        - Selenium
        - xlrd
        - xlwt
        - …..(For the detail Please see the requirement.txt in the project package or the APPENDIX A)
- Util
    - Ngrok

## 3.0 Deployment

1. Prerequisites
    1.1. Use current Google Account or Register a new one to log in the DialogFlow https://dialogflow.com/ → Click the Sign In or Go to Console

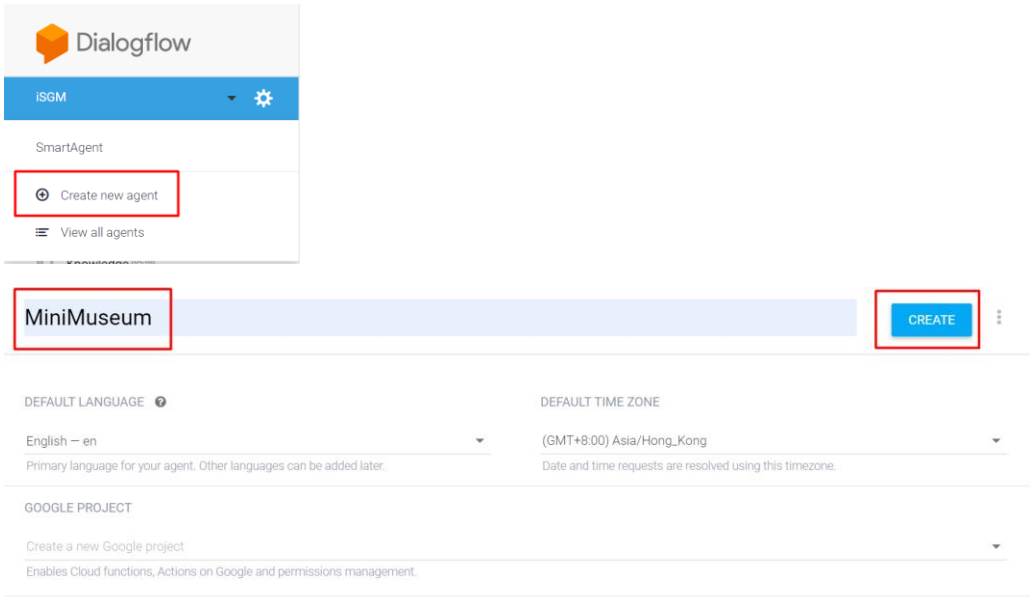



    1.2. Download the source code package of the project from the GitHub repo:

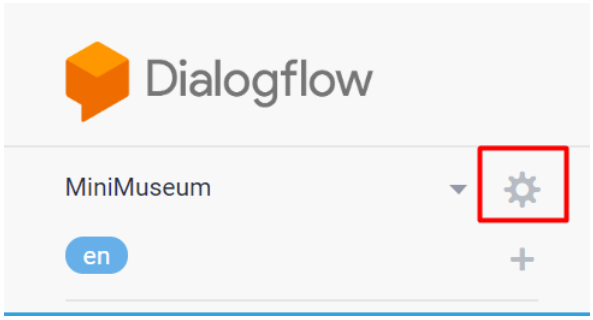
`git clone git@github.com:FoFxjc/IRS-CS-2019-08-31-IS01FT-GRP6-MiniMuseum-Chatbot.git`

    1.3. Download and Install the work on your machine from the ngrok official website https://ngrok.com/
    1.4. Install python3.7 on your machine (and use pip to install pipenv) https://www.python.org/
    1.5. Prepare a smartphone installed the Google Assistant and login with the same account used for the DialogFlow login

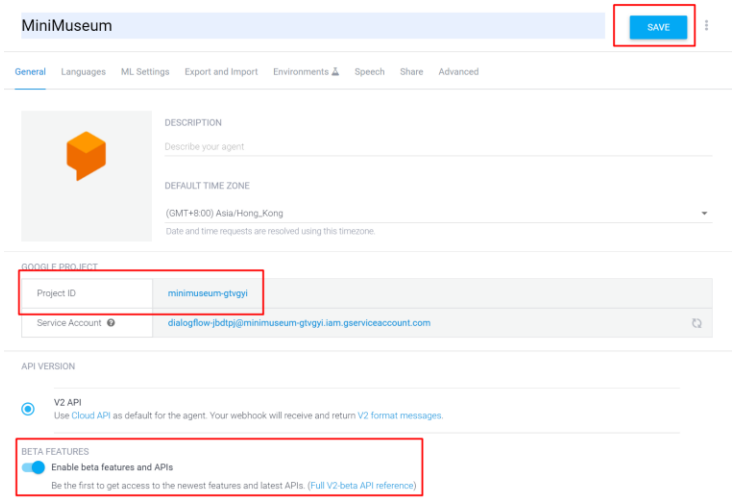
2. Set up DialogFlow
    2.1. Create a new agent called “Mini Museum”



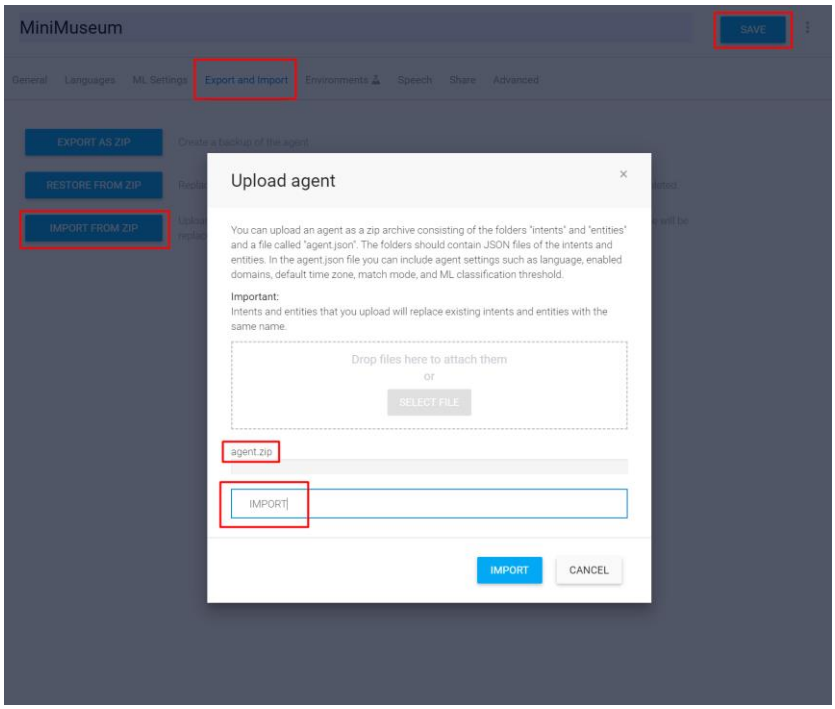

2.2. Click the setting button



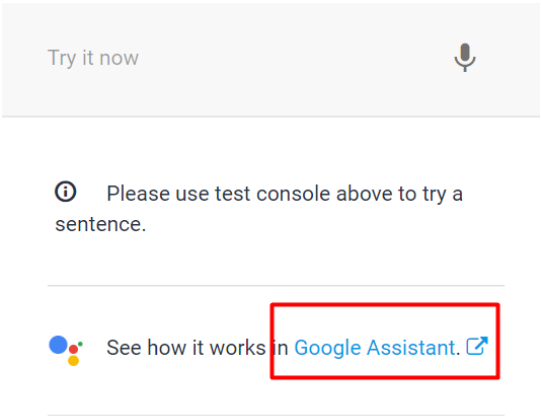

2.3. Record the project-id of the agent and open the Enable Beta Feature and APIs Switch, then remember Click Save Button



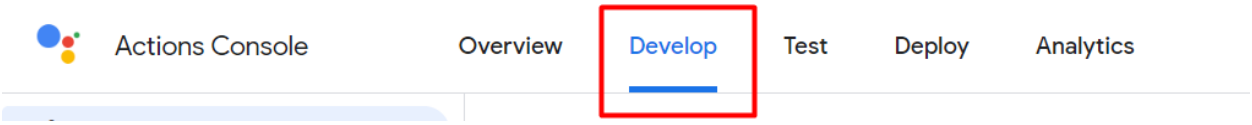
2.4. Import the zip file “MiniMuseumAgent.zip” (in the SystemCode Folder)



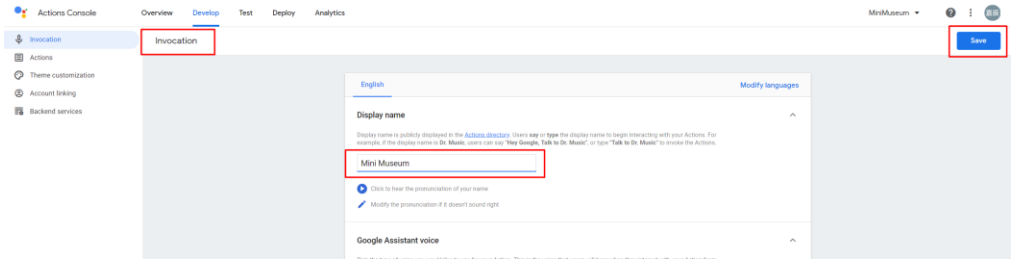

2.5. Click the Google Assistant on the right panel of the page

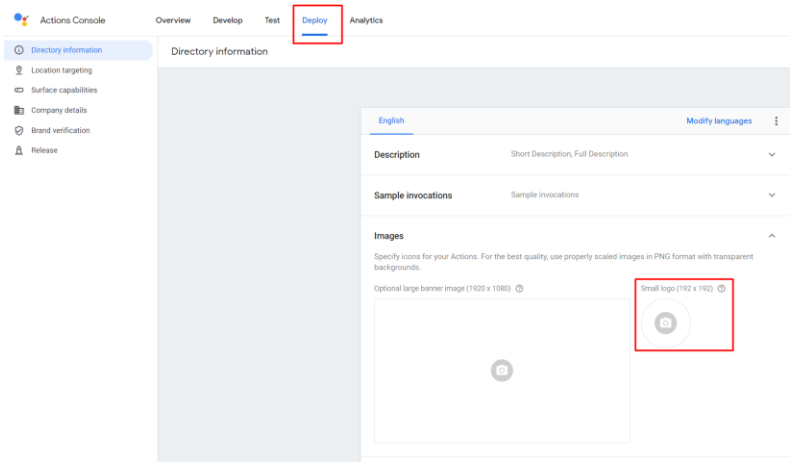



2.6. Click the Develop Button on the Top Menu Bar of the page




2.7. In the Invocation Page, input the Agent’s Display Name as “Mini Museum”, Then Click Save Button




3. Start backend server

The backend server code is in the "MiniMuseumBackend.zip" (in the SystemCode Folder)

3.1. Unzip the file

3.2. Use the pip or pipenv to install the requirements of the backend

   a. If you use pipenv, please use `pipenv install` in the root folder
   b. If you use pip and you want to install requirements into the original python environment, please use "pip install -r requirements.txt"

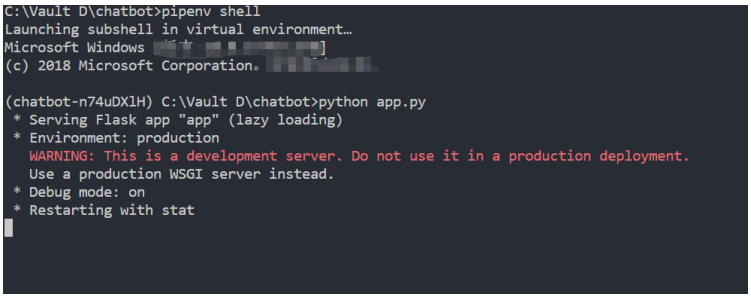
3.3. After the installation

   a. <u>Before starting the backend application, please modify the code in the app.py with the Agent project-id recorded before</u>. (This is an essential part of giving the capability to the backend for controlling the context in DialogFlow Conversation)

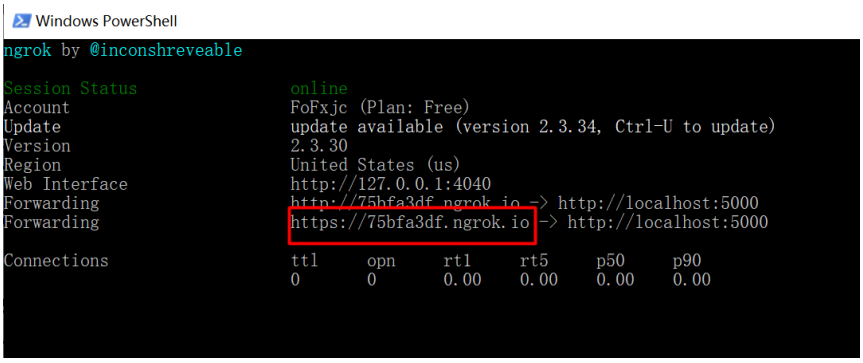
```
import logging
from flask import Flask
from flask_assistant import Assistant
from flask_assistant import ask, tell, event, build_item
from flask_assistant import context_manager
from IntentHandler.MuseumDetail import *
from IntentHandler.SupportDetail import *
from IntentHandler.RecommendActivity import *
from IntentHandler.ExhibitionDetail import *
from IntentHandler.ProgrammeDetail import *
from IntentHandler.DiningDetail import *
from IntentHandler.RetailDetail import *
from IntentHandler.utils.NameUtil import abbrName


app = Flask(__name__)
app.config['INTEGRATIONS'] = ['ACTIONS_ON_GOOGLE']
# Please use your agent's project_id
assist = Assistant(app, route='/', project_id="minimuseum-gtvgyi")
```

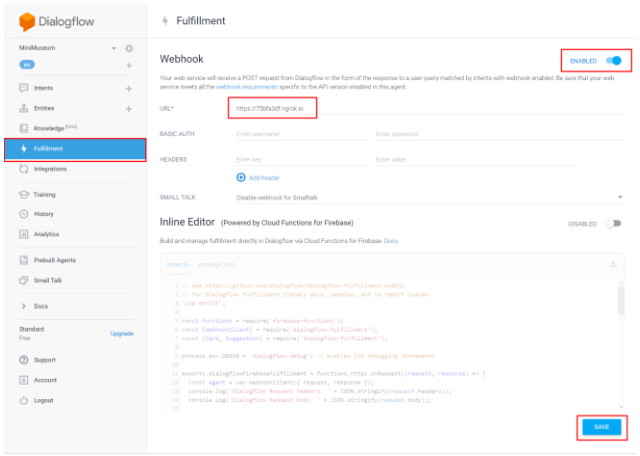
b. If you use pipenv to install, please use `pipenv shell` to activate the virtual environment. Then use `python app.py` in the root folder to start the Flask server.
c. If you use pip to install, please use the commend `python app.py` in the root folder to start the Flask server.
d. Our project's default port is 5000




3.4. Start the ngrok to explore the local IP and port




a. Run the ngrok
b. For Mac/Linux: ngork http 5000
c. For Windows: ngork.exe http 5000
d. Please copy the https URL in the command window
e. Paste this URL into the DialogFlow Fulfillment



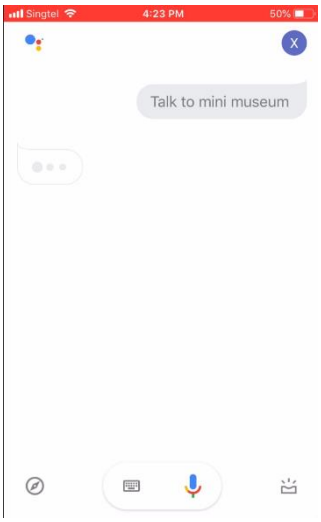
1) Click the Fulfillment Button
2) Paste the URL to the Webhook URL and Click the Save at the bottom of the Page
3) Make sure your have already Clicked the Save Button

## 4.0 Walkthrough

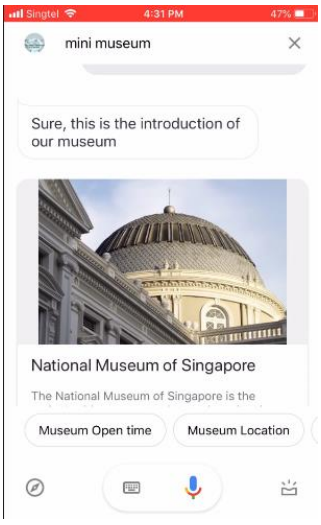
For the demonstration, we use the IOS phone installed the Google Assistant and log in the same account for logging in the DialogFlow

We recommend the Smart Device, like Android, IOS phone or tablet to do the test, because the Action on Google Simulator cannot provide the best test or user experience.

Please type “Talk to Mini Museum” in the Google Assistant or speak to the Google Assistant with the same sentence.





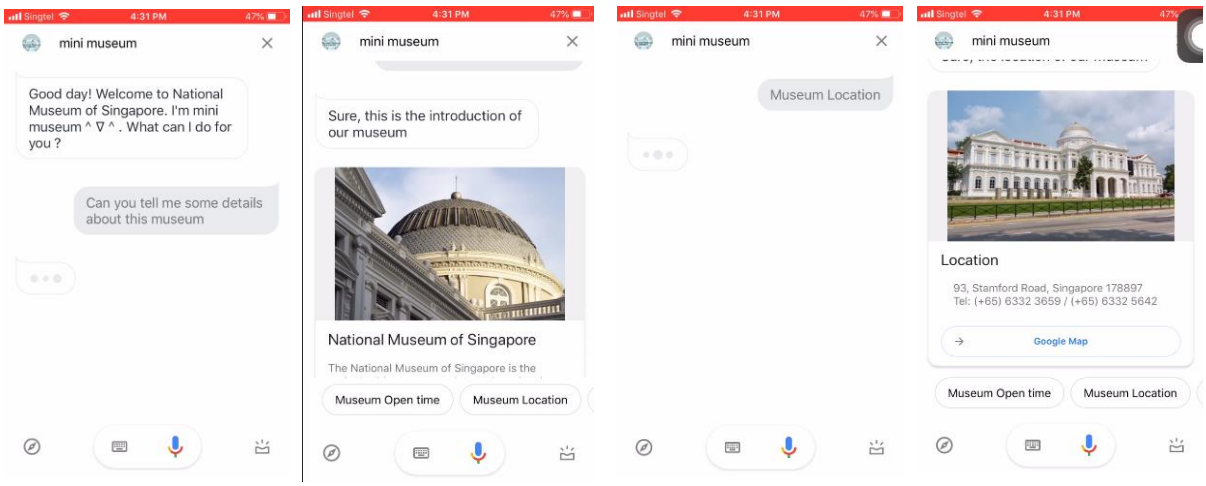


# 5.0 Business Scenarios – Sample Input and System Output

## 5.1 Scenario 1 Museum Info

| Scenario | Intents |
|---|---|
| For a tourist who is interested in the fundamental information of the National Museum of Singapore<br><br>**Sample Input**<br>1. Can you tell me some details about this museum?<br>2. Give me the location of this museum?<br>3. When does this museum open?<br>4. Tell me something about Group Visits<br>5. How much are the tickets of this Museum?<br>6. Can I take a vlog in the museum?<br>7. Call tell me something about the Amenities of this museum? | Mini Museum can provide the information on introduction, open time, location, ticket price, guided tours, group visits, accessibility, photography and Filming rules and venue rental<br><br>Intents:<br>MiniMuseum – MuseumInfo Group<br>• getMuseumDetail<br>  ○ inputContext:<br>  ○ outputContext: museumdetail<br>• getMuseumOpenTime<br>  ○ inputContext: museumdetail<br>  ○ outputContext: museumdetail<br>• getMuseumOpenTimeNC<br>• getMuseumLocation<br>  ○ inputContext: museumdetail<br>  ○ outputContext: museumdetail<br>• getMuseumLocationNC<br>• getMuseumTicketInfo<br>  ○ inputContext: museumdetail<br>  ○ outputContext: museumdetail<br>• getMuseumTicketInfoNC<br>• getMuseumGuidedTourInfo<br>• getMuseumGroupVisits<br>• getMuseumAmenitiesandEtiquetteInfo<br>• getMuseumVenueRentalInfo<br>• getMuseumAccessibilityInfo<br>• getMuseumPhotographyAndFilmingInfo<br><br>**# every Intent named ending with NC is to make sure the system can provide the same answer as using the normal one when the conversation is not in certain content** |

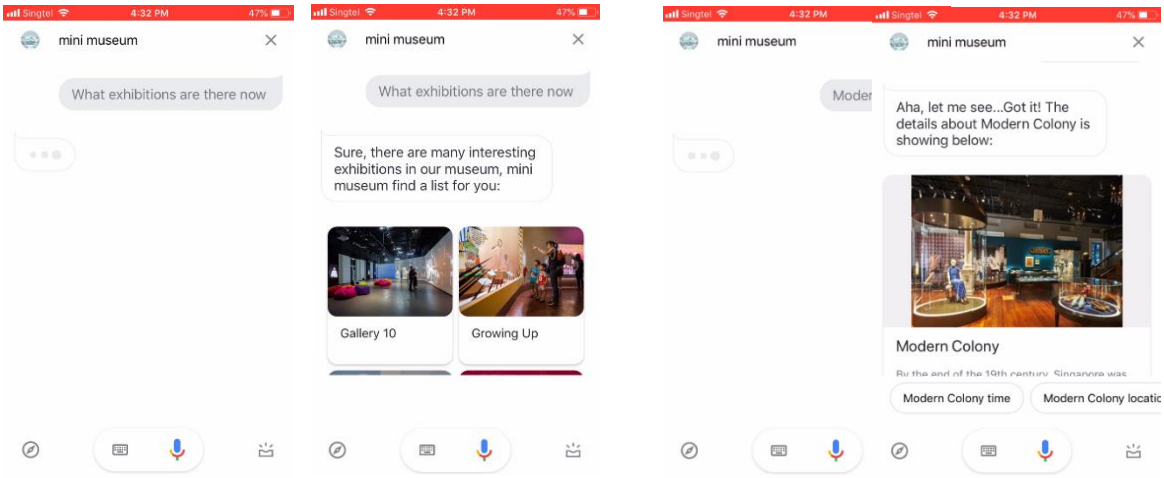
**Sample Screenshot:**



## 5.2 Scenario 2 Exhibition Info

| Scenario | Intents |
|---|---|
| For a tourist who is interested in the exhibition's information of the National Museum of Singapore<br>**Sample Input**<br>1. What exhibitions do you have now?<br>2. Gallery 10 details<br>3. Can you tell the location of Gallery 10<br>4. Does Gallery 10 require tickets? | Mini Museum can provide the information on exhibitions which are available about the location, open date, price and brief introduce<br><br>**Intents**<br>• getExhibitionIntro<br>• getExhibitionContent<br>o inputContext:<br>o outputContext: exhibitiondetail<br>o event: exhibition_tap<br>• getExhibitionTime<br>o inputContext: exhibitiondetail<br>o outputContext: exhibitiondetail<br>• getExhibitionTimeNC<br>o inputContext:<br>o outputContext: exhibitiondetail<br>• getExhibitionLocation<br>o inputContext: exhibitiondetail<br>o outputContext: exhibitiondetail<br>• getExhibitionLocationNC<br>o inputContext:<br>o outputContext: exhibitiondetail<br>• getExhibitionTicket<br>o inputContext: exhibitiondetail<br>o outputContext: exhibitiondetail<br>• getExhibitionTicketNC<br>o inputContext:<br>o outputContext: exhibitiondetail<br><br>**# every Intent named ending with NC is to make sure the system can provide the same answer as using the normal one when the conversation is not in certain content** |

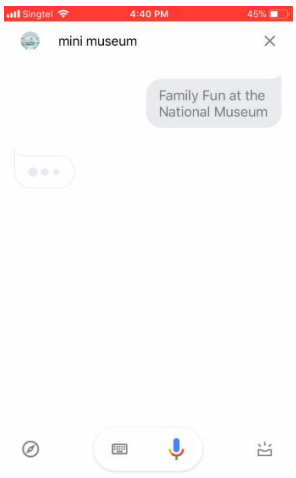
**Sample Screenshot:**

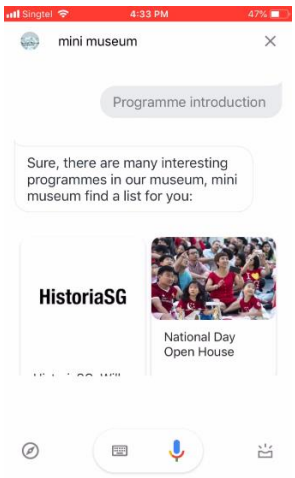


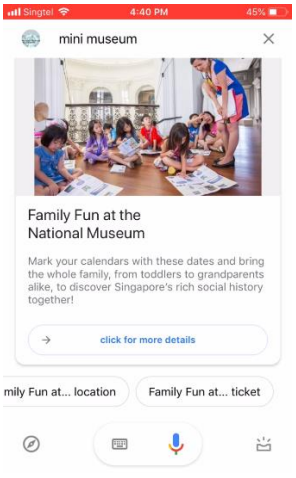
## 5.3 Scenario 3 Programme Info

| Scenario | Intents |
|---|---|
| For a tourist who is interested in the programmes information of the National Museum of Singapore<br><br>**Sample Input**<br>1. What programmes do you have now<br>2. I want to know more about school programmes.<br>3. What is the price for school programme<br>4. How much is the ticket for school programme | Mini Museum can provide the information on programmes which are available about the location, open date, price and brief introduce<br><br>**Intents**<br>• getProgrammeIntro<br>• getProgrammeContent<br>o inputContext:<br>o outputContext: programmedetail<br>o event: programme_tap<br>• getProgrammeTime<br>o inputContext: programmedetail<br>o outputContext: programmedetail<br>• getProgrammeTimeNC<br>o inputContext:<br>o outputContext: programmedetail<br>• getProgrammeLocation<br>o inputContext: programmedetail<br>o outputContext: programmedetail<br>• getProgrammeLocationNC<br>o inputContext:<br>o outputContext: programmedetail<br>• getProgrammeTicket<br>o inputContext: programmedetail<br>o outputContext: programmedetail<br>• getProgrammeTicketNC<br>o inputContext:<br>o outputContext: programmedetail<br><br>**# every Intent named ending with NC is to make sure the system can provide the same answer as using the normal one when the conversation is not in certain content** |

**Sample Screenshot:**



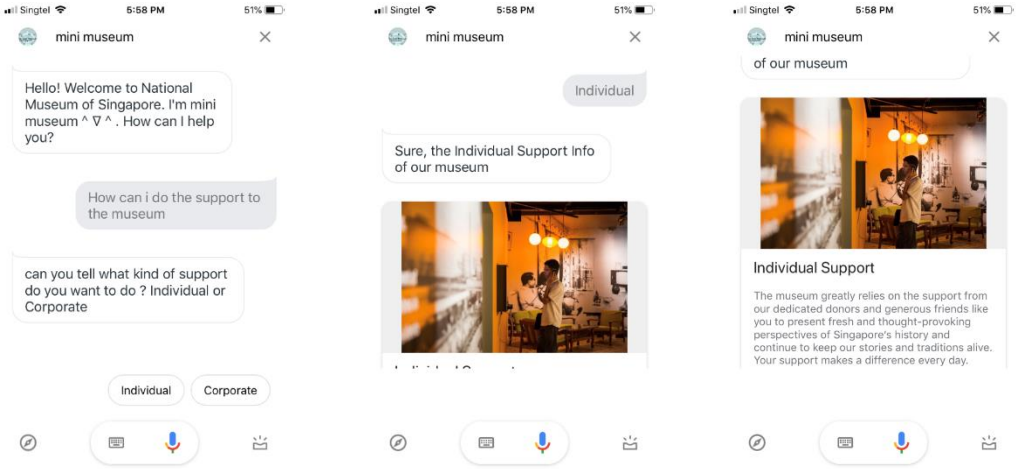



## 5.4 Scenario 4 Support Info

| Scenario | Intents |
|---|---|
| For a tourist who is interested in the support information of the National Museum of Singapore<br>**Sample Input**<br>**1.** How can I do the individual support to the museum<br>**2.** Can you give me some information about the volunteer of this museum | Mini Museum can provide the information of support about how to be a volunteer or a docent, how to do individual or corporate support, and the Fellowship of the Museum<br>**Intents**<br>• getSupportDetail<br>• getVolunteerInfo<br>• getFellowshipInfo |

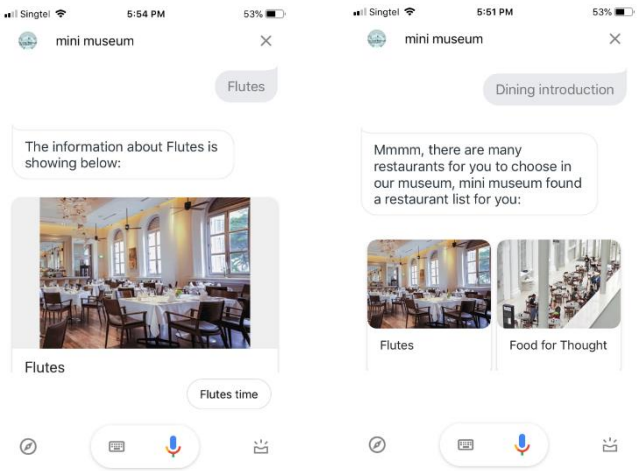
**Sample Screenshot:**



## 5.5 Scenario 5 Dining Info

| Scenario | Intents |
|---|---|
| For a tourist who is interested in the dining information of the National Museum of Singapore<br>**Sample Input**<br>1. Where can I have my lunch?<br>2. When does Flutes open? | Mini Museum can provide the information on dining about the location, available time and brief introduce<br><br>**Intents**<br>• getDiningIntro<br>• getDiningContent<br>  ○ inputContext: inputContext:<br>  ○ outputContext: diningdetail<br>  ○ event: dining_tap<br>• getDiningTime<br>  ○ inputContext: diningdetail<br>  ○ outputContext: diningdetail<br>• getDiningTimeNC<br>  ○ inputContext:<br>  ○ outputContext: diningdetail<br><br>**# every Intent named ending with NC is to make sure the system can provide the same answer as using the normal one when the conversation is not in certain content** |

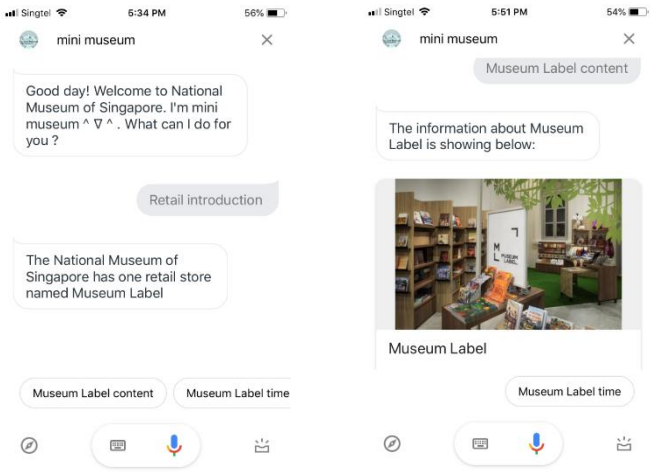
**Sample Screenshot:**



## 5.6 Scenario 6 Retail Info

| Scenario | Intents |
|---|---|
| For a tourist who is interested in the retail information of the National Museum of Singapore<br>**Sample Input**<br>1. where can I buy souvenirs<br>2. when does museum label open | Mini Museum can provide the information on retail about the location, available time and brief introduce<br><br>**Intents**<br>• getRetailIntro<br>• getRetailContent<br>  ○ inputContext:<br>  ○ outputContext: retaildetail<br>• getRetailTime<br>  ○ inputContext: retaildetail<br>  ○ outputContext: retaildetail<br>• getRetailTimeNC<br>  ○ inputContext:<br>  ○ outputContext: retaildetail<br><br>**# every Intent named ending with NC is to make sure the system can provide the same answer as using the normal one when the conversation is not in certain content** |

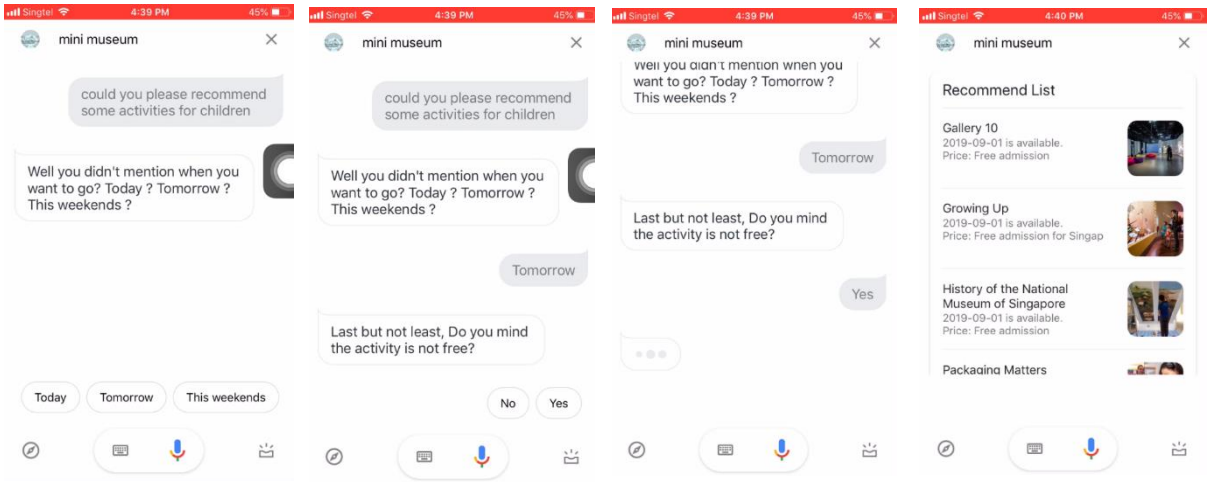
### Sample Screenshot:



## 5.7 Scenario 7 Recommend Info

| Scenario | Intents |
|---|---|
| For a tourist who wants to get some recommendation of his journey in the National Museum of Singapore<br>**Sample Iuput**<br>1. Can you recommend some activities from August 6th to August 20th?<br>2. Our family will be playing in Singapore from August 6th to August 16th. Can you recommend some activities that can be visited in the National Museum during this period?<br>3. Can you recommend some free exhibitions that can be visited for tomorrow?<br>4. Can you recommend me some activities in August? | Mini Museum will ask several questions to confirm the demands of his journey and provide the recommendation of exhibitions and programmes by the information got from tourist, e.g. the group type （For Family, Students）, the activity type (Workshop, Lectures), price and availability date.<br>**Intents**<br>• getRecommendInfo<br>  ○ inputContext:<br>  ○ outputContext: recommenddetail<br>• getRecommendInfo_GroupType<br>  ○ inputContext: recommenddetail<br>  ○ outputContext: recommenddetail<br>• getRecommendInfo_ActivityType<br>  ○ inputContext: recommenddetail<br>  ○ outputContext: recommenddetail<br>• getRecommendInfo_Date<br>  ○ inputContext: recommenddetail<br>  ○ outputContext: recommenddetail<br>• getRecommendInfo_DatePeriod<br>  ○ inputContext: recommenddetail<br>  ○ outputContext: recommenddetail<br>• getRecommendInfo_Price_No<br>  ○ inputContext: recommenddetail<br>  ○ outputContext: recommenddetail<br>• getRecommendInfo_Price_Yes<br>  ○ inputContext: recommenddetail<br>  ○ outputContext: recommenddetail |

**Sample Screenshot:**



# 6.0 Appendixes

**Appendix A**:
Backend Dependencies

- aniso8601==7.0.0
- certifi==2019.6.16
- chardet==3.0.4
- Click==7.0
- Flask==1.1.1
- Flask-Assistant==0.3.91
- idna==2.8
- itsdangerous==1.1.0
- Jinja2==2.10.1
- MarkupSafe==1.1.1
- requests==2.22.0
- ruamel.yaml==0.16.0
- ruamel.yaml.clib==0.1.0
- python-dateutil==2.8.0
- selenium==3.141.0
- urllib3==1.25.3
- Werkzeug==0.15.5
- xlwt==1.3.0
- xlrd==1.2.0
- pandas==0.25.0