Developer Guide

Overview

This is a developer guide for final year project titled **AI Chatbot for University Programming Courses**. This guide will walk the user through the different steps and processes to deploy the chatbot system implemented in this project. This guide will also highlight the file structure and the code related to each feature in the chatbot system. For better understanding of the overall architecture and algorithm involved in the system, please refer to the final report of this project.

Links to different chatbots & announcement channel

General Chatbot: https://t.me/source_chat_test_bot

Staff Chatbot: https://t.me/source_chat_staff_bot

Announcement Channel: https://t.me/joinchat/AAAAAEo1n0cdfzebUWk--w

Contents

Getting Started

Signing up for Google Cloud & Dialogflow Account

Google Cloud Account Creation: https://console.cloud.google.com/freetrial

Dialogflow Account Creation: https://dialogflow.cloud.google.com/

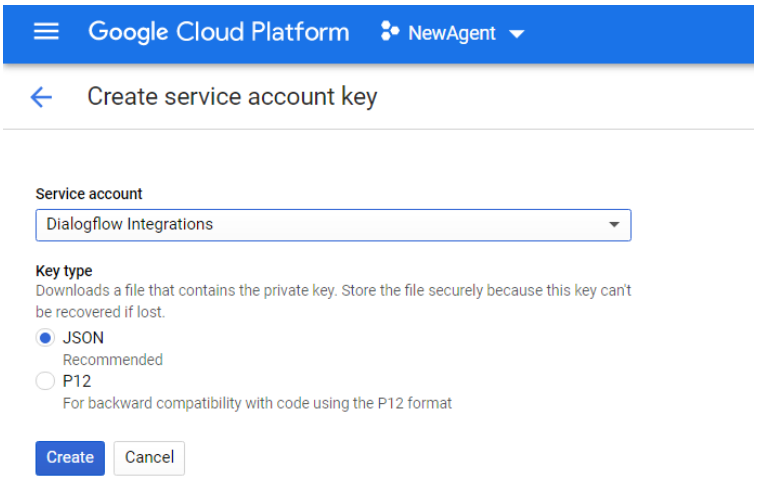
Basic Concepts & Understanding of Dialogflow

1. https://cloud.google.com/dialogflow/docs/basics
2. https://towardsdatascience.com/understanding-the-basic-conversational-ai-concepts-with-dialogflow-b0604d957d5c

Setting up of Dialogflow Project

Adapted from: https://cloud.google.com/dialogflow/docs/quick/setup

1. Create a Cloud Platform Project: https://console.cloud.google.com/project
2. Enable billing for project: https://support.google.com/cloud/answer/6293499#enable-billing
3. Enable the Google Cloud Dialogflow API: https://console.cloud.google.com/flows/enableapi?apiid=dialogflow.googleapis.com
4. Set up authentication with a service account so you can access the API from your local workstation: https://cloud.google.com/docs/authentication/getting-started
5. Ensure that the service account is set to Dialogflow Integrations and that the service account key is downloaded in json format, this file will be called the **Dialogflow json agent file**.



Building and Setting up of Dialogflow Agent

Adapted from: https://cloud.google.com/dialogflow/docs/quick/build-agent

This section will present the steps to deploy an agent on Dialogflow. Head to https://console.dialogflow.com/ to create a chatbot agent as seen from the figure below. The agent will require a name before it is created. Follow default settings for agent creation as seen in Figure 1.

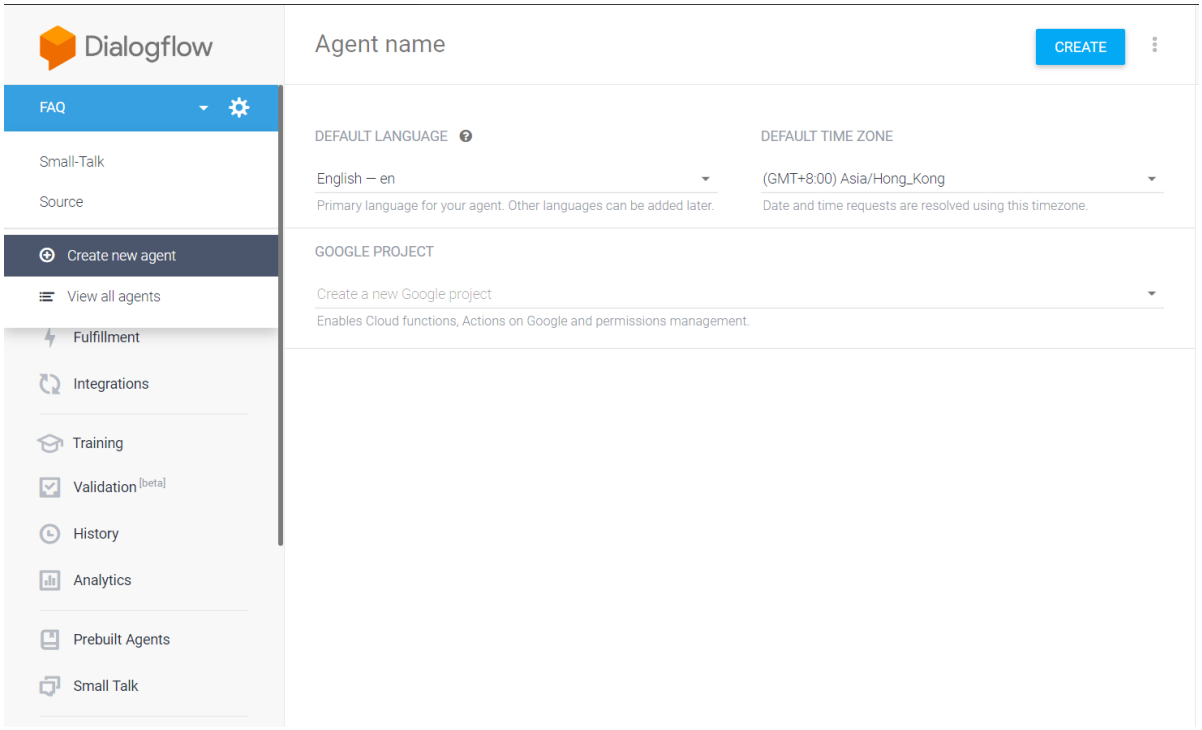


Figure 1: Dialogflow Interface to create Agent

Configuration of settings for Dialogflow Agent

Head to the **general settings** of the agent to obtain the necessary API tokens, Project ID and Service Account ID. Ensure that the settings for the API version are also set to the correct version with Beta Features enabled as seen in Figure 2. Next head on to **ML settings** to ensure that the ML classification threshold is set at 0.3 and hybrid is set for the match mode. Exclusive of the ML classification threshold and the match mode, ensure that the rest of the settings are set to the default configurations as seen in Figure 3.

The ML settings for ML classification threshold and match mode can be tweaked based on the preference of the developer. Developer should take notice of the trend and confidence levels return in the chat log of the overall chat system.

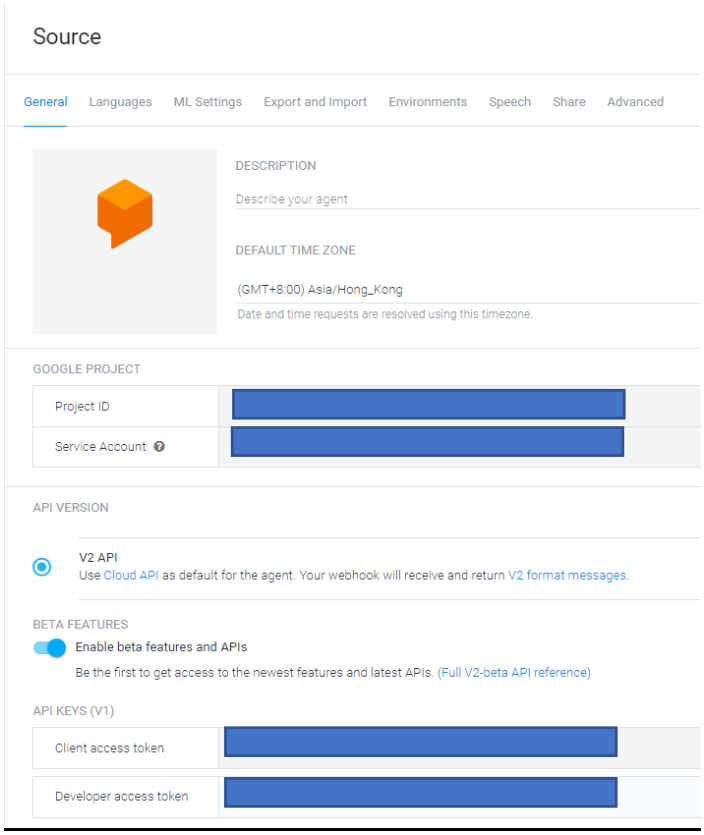


Figure 2: General Settings of Dialogflow Agent

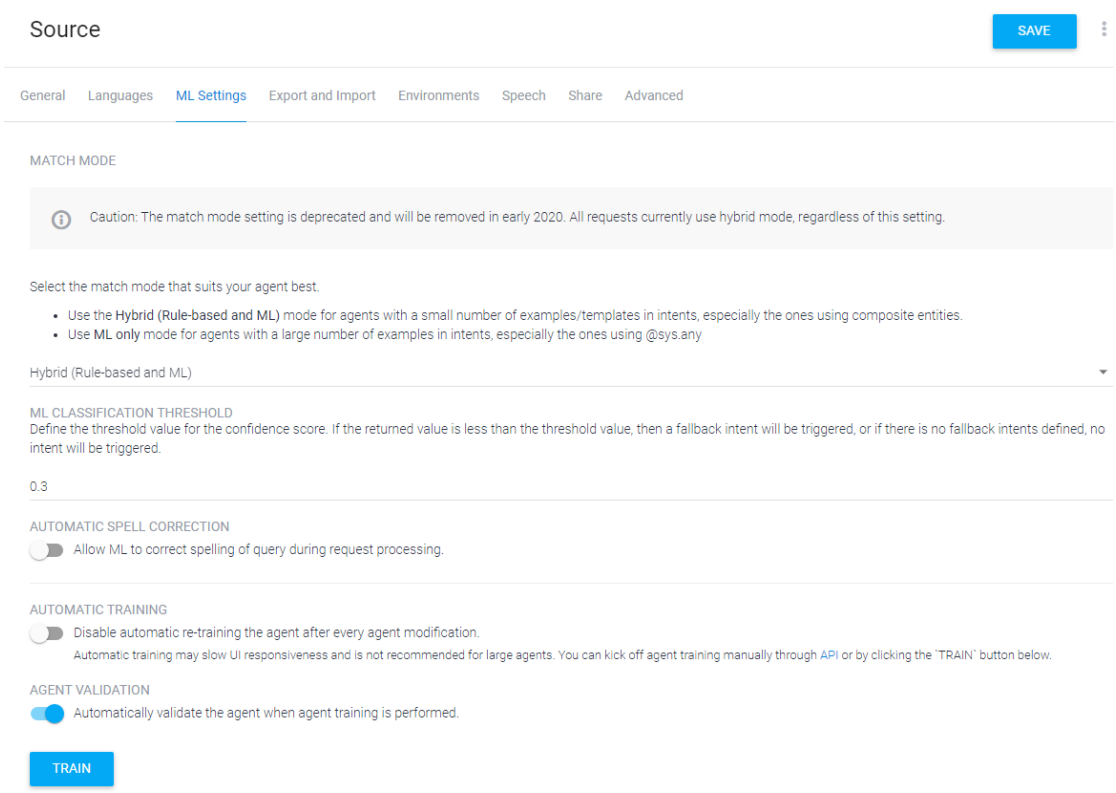


Figure 3: ML Settings of Dialogflow Agent

Creating an Intent in Dialogflow Agent

Once the Agent is ready, questions and answers can be added into the chatbot. This section showcases how to add one set of question and answer to the chatbot. Firstly, navigate to the intents section of the agent. At the top right of the Dialogflow interface, select create intent. On the create intent page, the developer is required to insert the intent (also known as the title) for the set of question and answer. In this tutorial, the intent is named as “source.codestyle”.

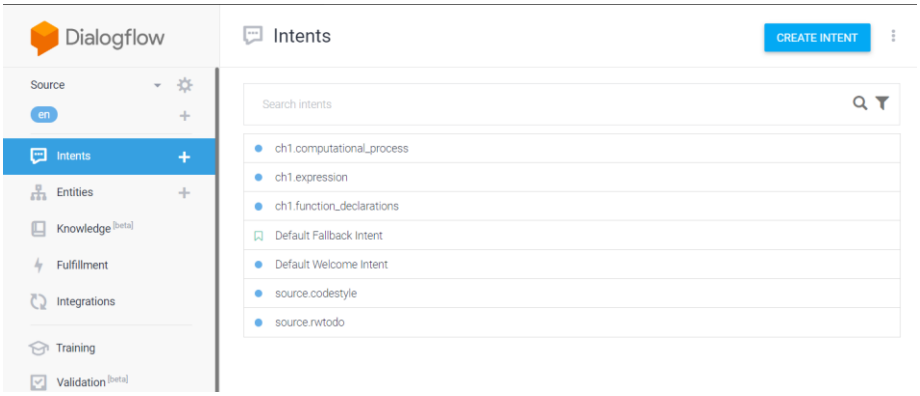


Fig 4: Dialogflow Interface of Intent Section

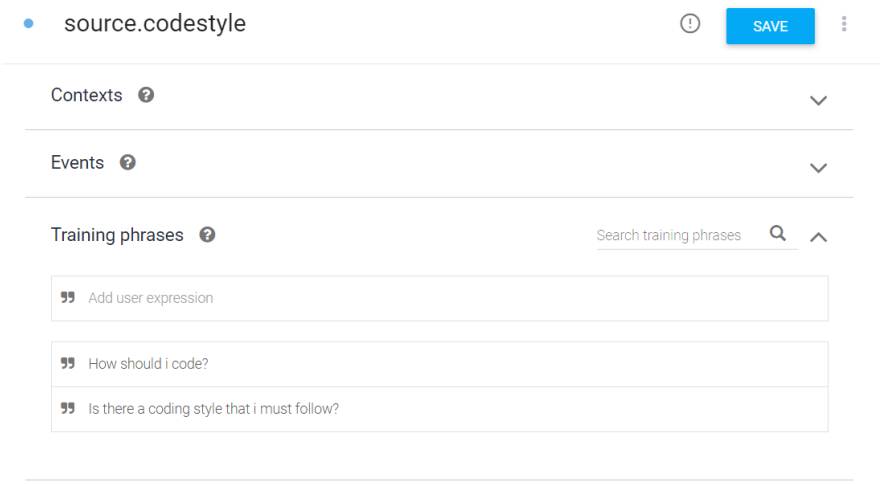


Fig 5: Dialogflow Interface to create Intent and insert training phrases

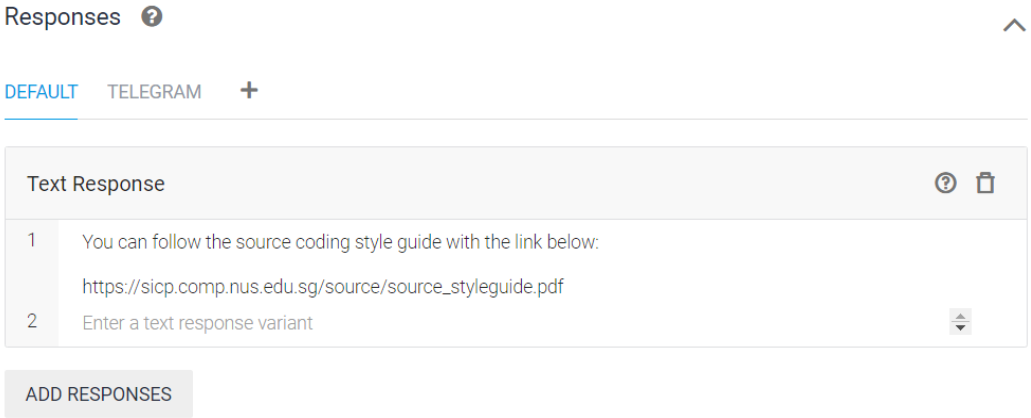


Fig 6: Dialogflow Interface to create Intent and insert responses

In an intent, training phrases and responses can be added, once all training phrases and responses are added, it is required for the developer to save the intent at the top right section of the Intent Interface. Training phrases are defined as questions which are used to train the chatbot to interpret a user request, in the figure, a few questions are crafted such as 'How should I code?' and 'Is there a coding style I must follow?'. Responses are defined as the response that developers have determined how the chatbot should respond to its users. Once the intent is saved, Dialogflow NLU engine will process the training phrases and extract the necessary keywords so that when a user types a request in the future, it will be able to process the request and determine whether it is the request matches the questions in its stored database. The result should be like the figure 8 below when a user inputs a request of 'how to code?' to the chatbot.

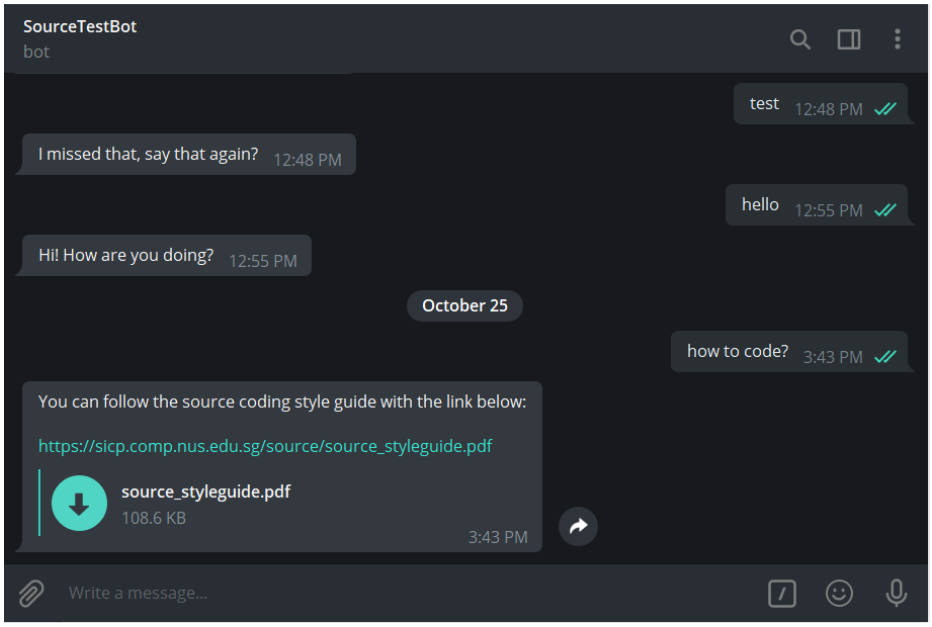


Fig 7: Results of request and response on prototype A.I. Chatbot

Enabling Read Write Access for Dialogflow Agent & Project

1. Obtain credentials from settings of Dialogflow agent
2. Enabling of API: https://console.cloud.google.com/flows/enableapi?apiid=dialogflow.googleapis.com
3. Granting Roles to Service Accounts: https://cloud.google.com/iam/docs/granting-roles-to-service-accounts
4. Granting Access Control for Dialogflow Agents: https://cloud.google.com/dialogflow/docs/access-control#gcpconsole
5. Ensure that editor role is given to Dialogflow agent and account

Requesting & Obtaining of Telegram API Key

To deploy a chatbot on Telegram, the developer is required to request a Telegram API key from the BotFather on Telegram. Figure 10 displays the steps to obtain the Telegram API key from BotFather. Once obtained, the API key can be inserted into the credentials python file of the main project folder.

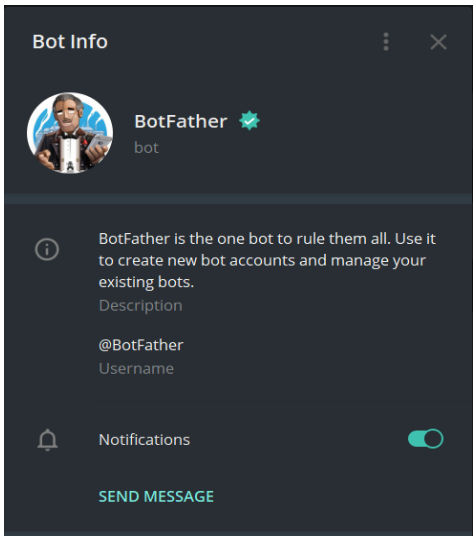


Fig 8: Description of BotFather on Telegram Platform

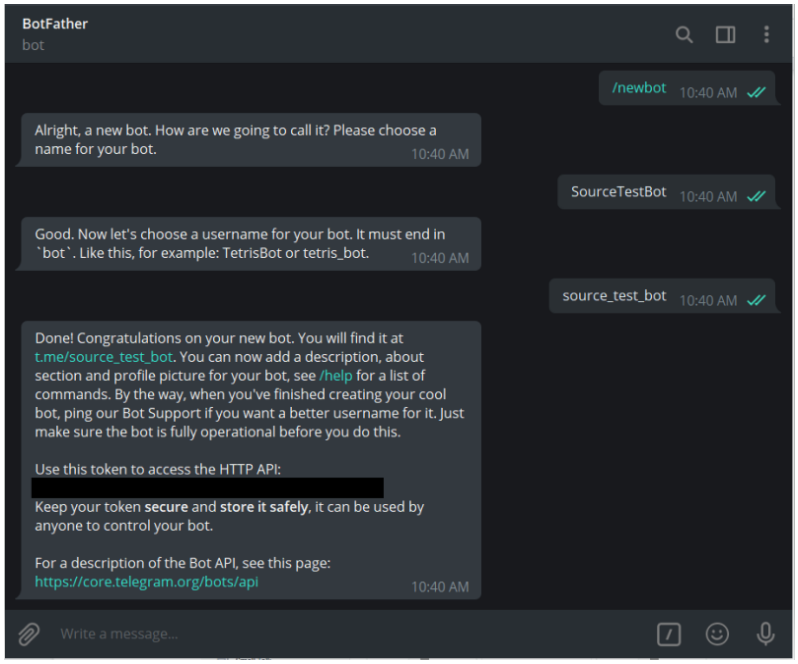


Fig 9: Steps to request API Key from BotFather on Telegram Platform

Setting up of Telegram Announcement Channel

Follow this link: https://telegram.org/faq_channels

Deployment

Installation

Pre-requisite:

Please ensure that Python 3 is installed in your computer running the program.

In the main project folder, there will be a file named **requirements.txt**. To install all related packages to the project, user can type:

pip install requirements.txt

File Structure

All files exist in the same folder.

Main Files

- sourcechat.py – General Chatbot Program
- adminchat.py – Staff Chatbot Program
- adminCredentials.py – Configuration file for Staff Chatbot Program
- generalCredentials.py – Configuration file for General Chatbot Program
- NewAgent-e6b62ce69677.json – Dialogflow Agent Json File
- requirements.txt – List containing Python Packages to be installed

Log Files

- sourceChatLog.json – General Chatbot Log storing user message, response and confidence level
- ticketlog.json –Ticket Log storing user messages and user IDs related to user messages
- ratingLog.json – Rating Log storing ratings of response to user questions
- deletedLog.json – Log that stores all deleted question answer pairs from rating system

Additional Programs

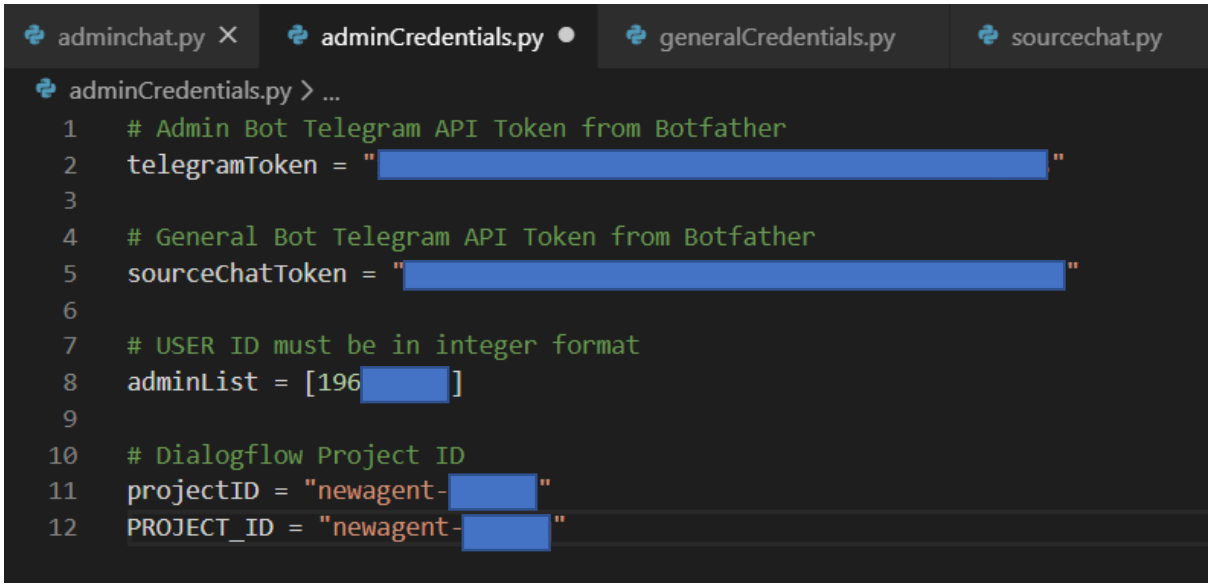
- uploadIntent.py – Additional script to upload question answer pairs in csv format
- generateResults.py – Additional script to generate comments added into responses from conversations between users and General Chatbot

Configuration

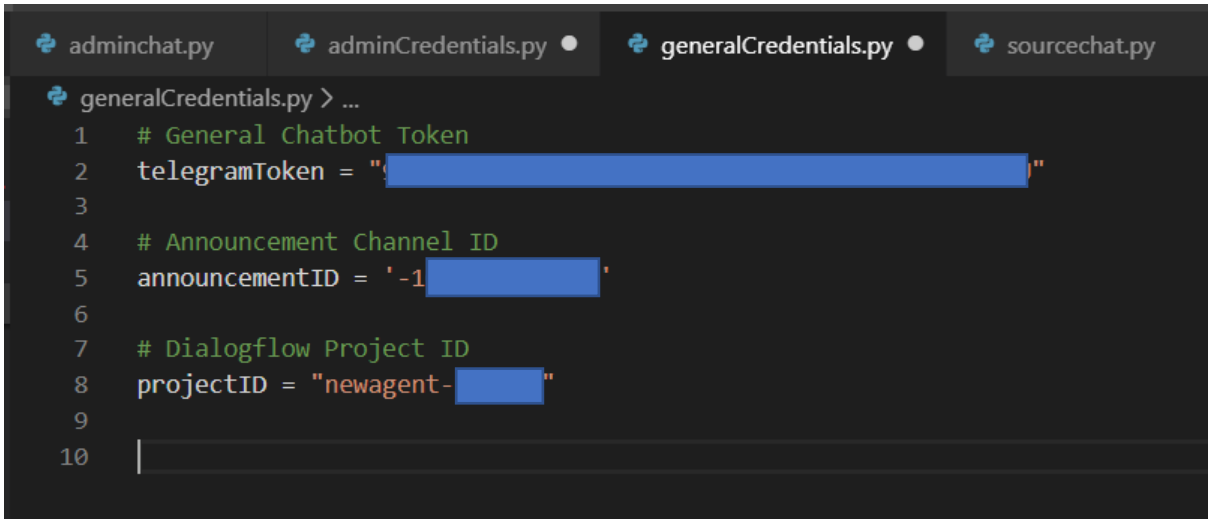
Configuration for Staff Chatbot

Things to note:

- USER ID added in adminList must be in integer format.



Configuration for General Chatbot



Execution of Chatbot System Program

1. Ensure that Dialogflow json agent file is in same project folder as chatbot program
2. This is a must do before running any of the chatbots:
   set GOOGLE_APPLICATION_CREDENTIALS=**NewAgent-e6b62ce69677.json**
   where the variable should be changed according to the name of the Dialogflow json agent file.
3. To run general chatbot program: **python sourcechat.py**
4. To run staff chatbot program: **python adminchat.py**

Additional Programs

Batch Upload of Question Answer Pair Datasets

1. Edit csv filename in script **uploadIntent.py**
2. set GOOGLE_APPLICATION_CREDENTIALS=NewAgent-e6b62ce69677.json
   where the variable should be changed according to the name of the Dialogflow json agent file.
3. python uploadIntent.py

An example of how a csv file containing the dataset should look like where Column A is questions while Column B is the answers to the different questions for each row.

	A	B
1	How do we define a variable of a value?	You can use let instead of const, however it is not available un
2	How should I name my functions?	In the spirit of communicating computational processes, good
3	how to determine the space taken for iteration and recursion process?	Every function and operation (x*y) occupies space. In the case
4	Is there any difference between an "iterative" function and a "tail-recur	Yes, they mean the same thing in Source. We make a distinctio
5	Where can I find learning materials for source academy?	https://sourceacademy.nus.edu.sg/material
6	What is the number of steps for Iterative function of fibonacci	O(n). In the reflection we would say k * n, but fundamentally it
7	Where can I find the specifications for Source?	The specification of Source in https://sicp.comp.nus.edu.sg/source/ and the lectures are your definitive resource on this. The implementation of Source may currently be a bit more generous, and may actually give "correct" results, meaning, results that are consistent with the JavaScript specifications. But please do not rely on this. Treat it as an "undocumented feature". Next time when we work on the Source implementation, we may decide to close this gap, and your program may stop working as expected.

Generate Comments from Chat Log

1. To run file: python generateResults.py

Design: Architecture

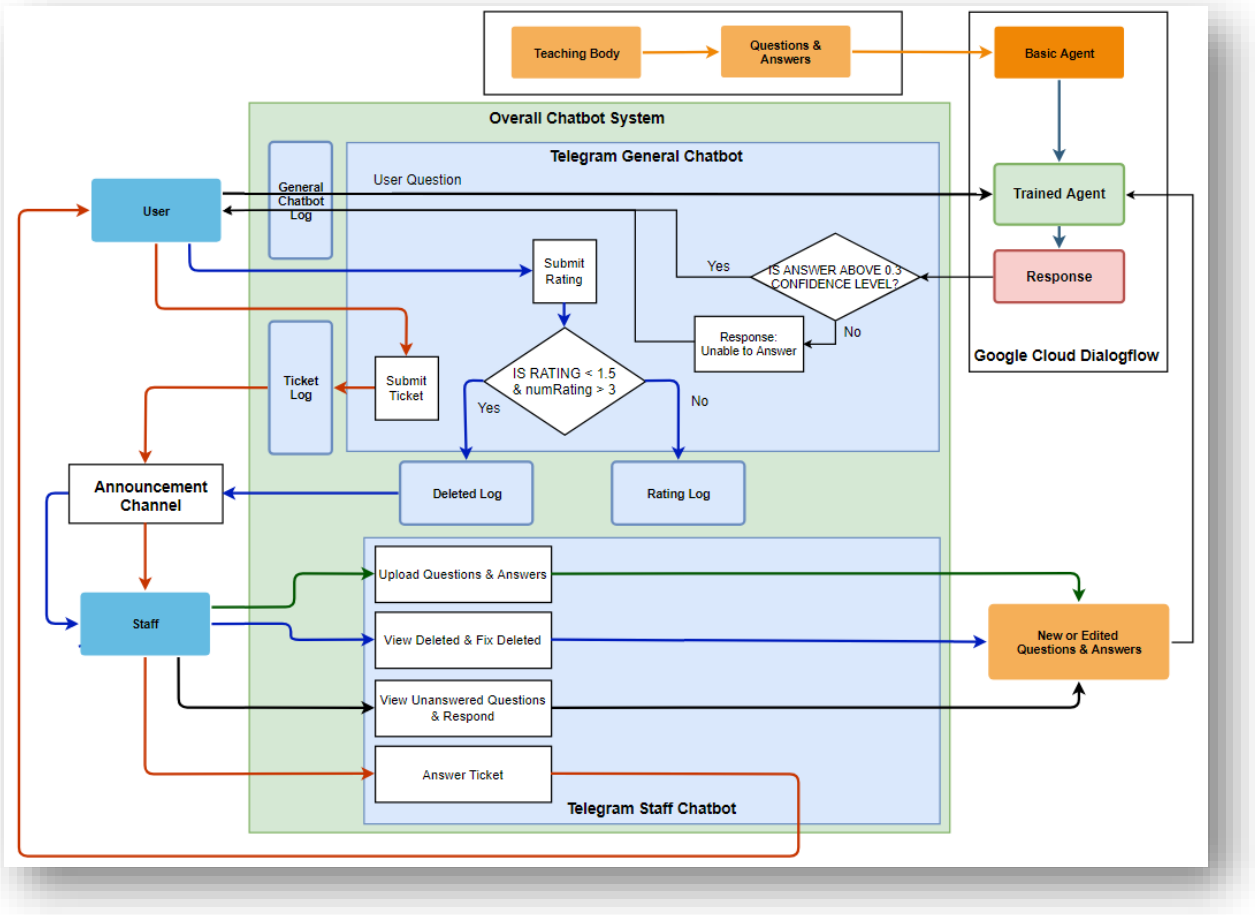


Fig A: Overall Architecture of Chatbot System

The general chatbot that the students will be able to interact with has **3 main features**, sending text questions to the trained agent, rating the answers given by the chatbot and submitting an urgent ticket for questions that do not have a desired answer. All user questions and Dialogflow Agent responses are logged to help facilitate the overall functions of the two chatbots. There is an announcement channel that notifies staff and the teaching body of submitted tickets and responses with bad ratings. The Staff Chatbot has **4 main features** that allows the teaching body to carry out follow up actions on the tickets and questions that are asked by the users. The features are as follows, allowing staff to view upload questions and answers that are categorized as text-based **"General Programming Concepts" questions**, allowing staff to view and respond to user tickets, allowing staff to view unanswered questions and allowing staff to view deleted answers via the rating system.

Implementation

This is the main function and structure of a chatbot program where the commands of the Telegram Chatbot are each instantiated and added using the CommandHandler class with a string representing the function and the function of the command. ConversationHandler is also a class which adds a command to the Telegram bot, however, it involves the use of a group of commands.

```
def main():
    """Start the bot."""
    updater = Updater(telegramToken, use_context=True)

    # Get the dispatcher to register handlers
    dp = updater.dispatcher

    # on different commands - answer in Telegram
    dp.add_handler(CommandHandler("start", start))
    dp.add_handler(CommandHandler("help", help))
    dp.add_handler(CommandHandler("ticket", ticket))
    dp.add_handler(CommandHandler("feedback", feedback))
    conv_handler = ConversationHandler(
        entry_points=[CommandHandler('rate', rate)],
        states={
            RATING: [MessageHandler(Filters.text, receive_rating)]
        },
        fallbacks=[CommandHandler('cancel', cancel)]
    )
    dp.add_handler(conv_handler)

    dp.add_handler(MessageHandler(Filters.text, textMessage))

    # log all errors
    dp.add_error_handler(error)

    # Start Telegram Bot
    updater.start_polling()
    print('Bot Started')

    # Run the bot until you press Ctrl-C or the process receives SIGINT,
SIGTERM or SIGABRT.
    updater.idle()

if __name__ == '__main__':
    startupCheck()
    main()
```

Code snippet of main function of General Chatbot Program

General Chatbot

Files involved: **sourcechat.py** and **generalCredentials.py**

Feature 1: Sending of text questions to Dialogflow agent

Functions involved: **textMessage, detect_intent_texts**

For this feature, user messages are received via the textMessage function and sends a request to Dialogflow agent by calling the function detect_intent_texts. A response with the confidence level and intent name is returned from the Dialogflow agent which can then be sent back to the user who sent the message. For each user message and response, a unique hash identity is generated and an object containing the confidence level, user message, response and intent name of the response is created with the unique hash identity as the key. The object is then stored in a json file also known as sourceChatLog.json.

Feature 2: Rating of the answers given by the General Chatbot

Functions involved: **rate, receive_rating**

For this feature, user will type the command /rate to an answer given by the General Chatbot, calling the rate function where the General Chatbot will prompt user to give a rating from 1 to 5. Once the rating is inputted, function receive_rating will be called, and rating will be saved in ratinglog.json. The function receive_rating will also be checked if the intent has been rated x number of times (controlled by variable thresholdNum) and that the threshold rating is below a certain value (controlled by variable ratingDeleteThreshold), the function will proceed to delete the question and answer pair from the Dialogflow agent and store in the Deleted Log which is known as deletedLog.json. The function will then proceed to make an announcement to the announcement channel to notify the teaching staff of the poorly rated answer and deleted question answer pair.

Feature 3: Submitting of urgent ticket with question

Functions involved: **ticket**

For this feature, user will type /ticket to an answer given by the General Chatbot, this will call the ticket function where the General Chatbot will store the user message and user question in a ticket log and make an announcement in the announcement channel stating that the question asked currently is urgent and requires attention from the staff.

Additional Feature 1: Comment on answer given by General Chatbot

Functions involved: **feedback**

For this feature, user types /feedback and can type in comments to the answer, giving feedback to staff on how to improve the answer given by the staff. Comment received from the command will be included into the sourceChatLog.json for the unique key.

Staff can then use generateResults.py script to generate a CSV file that shows the relevant details that help the staff to see how they can improve the answer and see which answer requires improvement.

Staff Chatbot

Files involved: **adminchat.py** and **adminCredentials.py**

Feature 1: Uploading of questions and answers to Dialogflow agent

Functions involved: **startUpload, receiveQuestion, receiveAnswer, uploadQuestion**

For this feature, staff user types /uploadquestion, Staff Chatbot then prompts staff to insert question and then prompts user to insert answer. Once both question and answer are inserted, a confirmation is required by the staff user before it gets uploaded. Once confirmation is given, question and answer are uploaded to Dialogflow agent.

Feature 2: View and respond to user tickets

Functions involved: **generateTickets, selectTicket, sendAnswer**

For this feature, staff user types /gettickets, Staff Chatbot will then show a list of available tickets. Staff user is prompted to select a ticket. Once ticket is selected, staff user is prompted to return an answer back to Staff Chatbot. Once answer is inputted, answer is sent back to user via Telegram. Ticket is also removed from ticket log.

Feature 3: View unanswered questions

Functions involved: **generateUnanswered, chooseUnans, deleteUnans**

For this feature, staff user types /getunanswered, Staff Chatbot lists all user messages that are unable to be interpreted. Staff user has the option to remove the unanswered user messages from the list. Staff user can then answer unanswered user messages by uploading the questions using Feature 1 of the Staff Chatbot.

Feature 4: View deleted answers via the rating system.

Functions involved: **viewdelete, selectDeleted, receiveOption, uploadDeletedQuestion, receiveEditted, uploadEditted**

For this feature, staff user types /viewdelete, the Staff Chatbot displays a list of deleted question and answer pairs, user is then required to select an option form the list. Once an option is selected, user can view the full details of the deleted question and answer with the ratings from other users and their user messages for that specific intent. Staff user is given the option to reupload the question and answer pair or edit the answer to the question. Once an option is selected, the question and answer (adjusted or not adjusted) is uploaded to the Dialogflow agent. Staff user can also gain additional insights from user messages and upload new question and answer pairs to the Dialogflow agent.

Troubleshooting

- Setting up service key in OS environment: https://cloud.google.com/dialogflow/docs/quick/setup#auth-env
- Ensuring that Dialogflow API works: https://cloud.google.com/dialogflow/docs/quick/api
- Relevant Python Github Repositories:
    - https://github.com/googleapis/dialogflow-python-client-v2
    - https://github.com/python-telegram-bot/python-telegram-bot
- Obtaining channel id of announcement channel: https://stackoverflow.com/a/33862907