User Manual:

## Public Transport Chatbot

## 3rd Year Project

## Michael O’Hara 16414554

## Matthew Nolan 16425716

## Supervisor: Prof. Gareth Jones

# Table of Contents:

# Joining the Chatbot

1.1 - How to join the chatbot

# Small talk with the Chatbot

* 1. – Interacting with the chatbot

# Help Function

3.1 – Getting Help

# Google integration

4.1 – Google within the Chatbot

# Querying the Chatbot

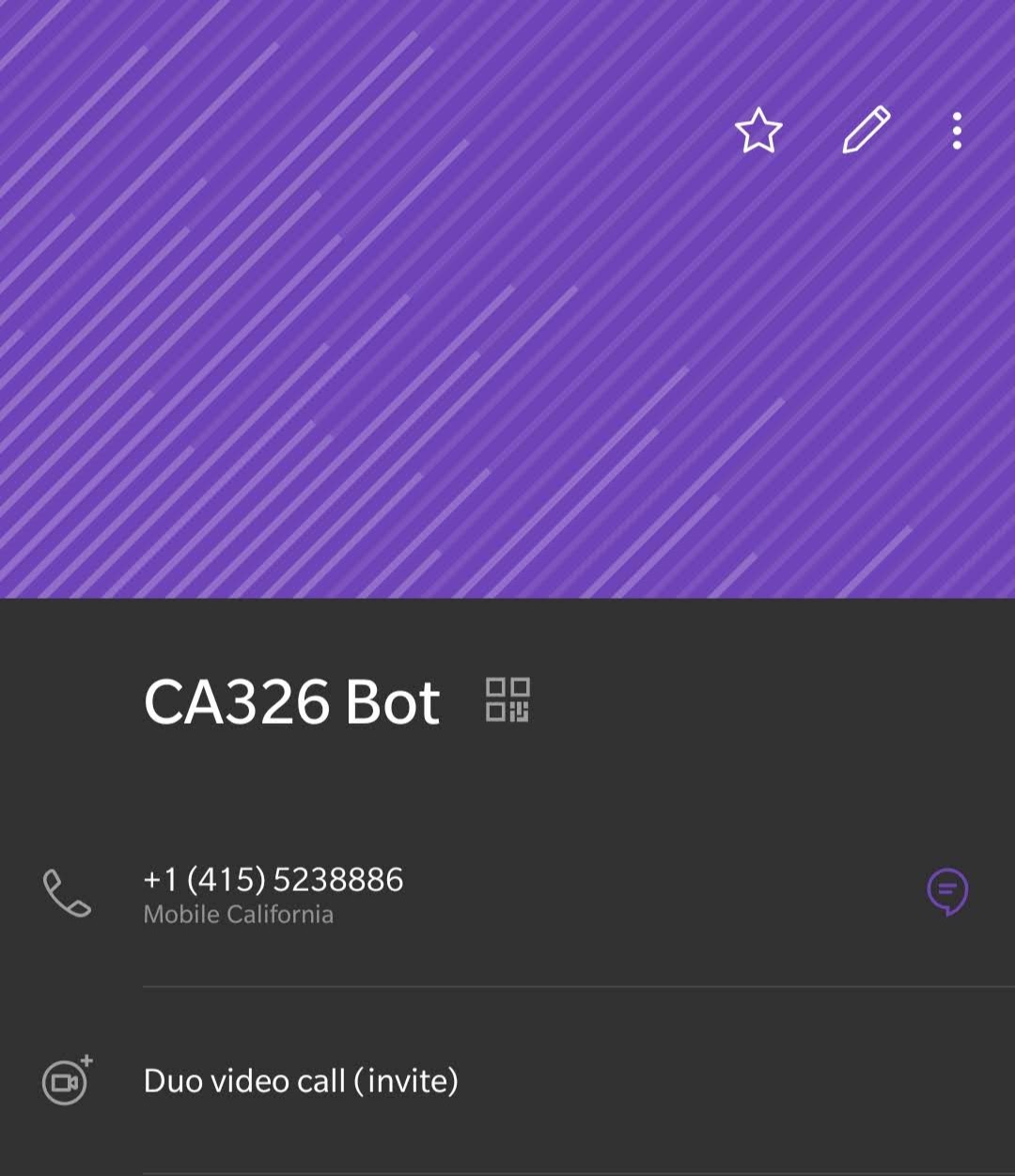
5.1 – How to information about the trains

# Section 1 – Joining the sandbox

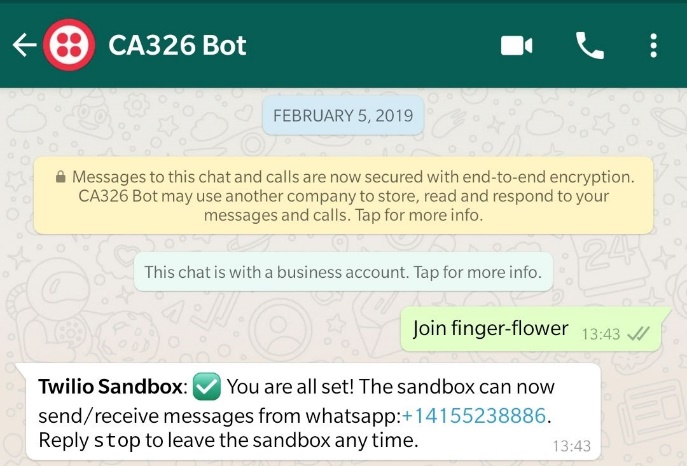
### 1.1 – How to join the chatbot

The user must have WhatsApp installed on their phone and set up with their phone number. This should take no more than 5 minutes.

Once that is done the only steps required by the user is to add the bots contact details to their phonebook. The name of the bot can be put in as anything (CA326 Bot for example as shown below) but the number must be the one shown below ( +1 (415) 523 8886).



Once the user has WhatsApp set and the contact added to their phone then the user has to simply open a chat with the contact on WhatsApp and type in the join phrase in order to join the WhatsApp sandbox that the chat bot operates out of. The sandbox phrase we are using is “join finger-flower”

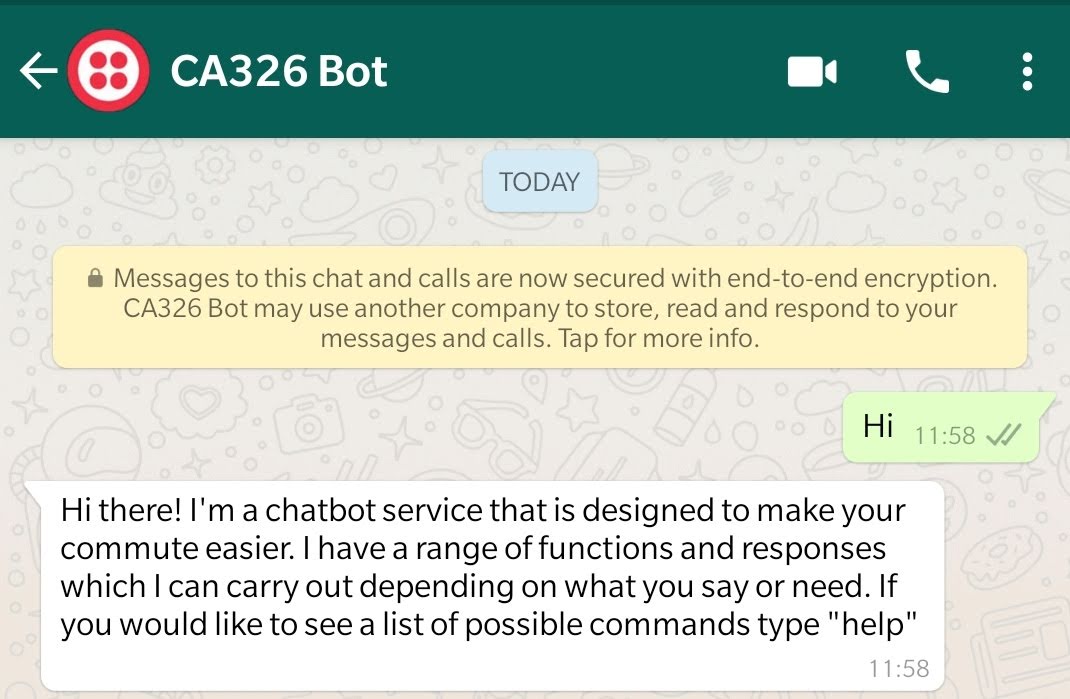


This allows our sandbox to remember your number and thus be able to send/receive messages to and from the sandbox. Once you receive a message like the one above you should be good to go!. That is all the steps the user needs to take in order to get access to the chatbot service. The bot will only reply when prompted first by the user meaning there is no need to worry about spam messages coming from it.

# Section 2 – Small Talk

### 2.1 – Interacting with the chatbot

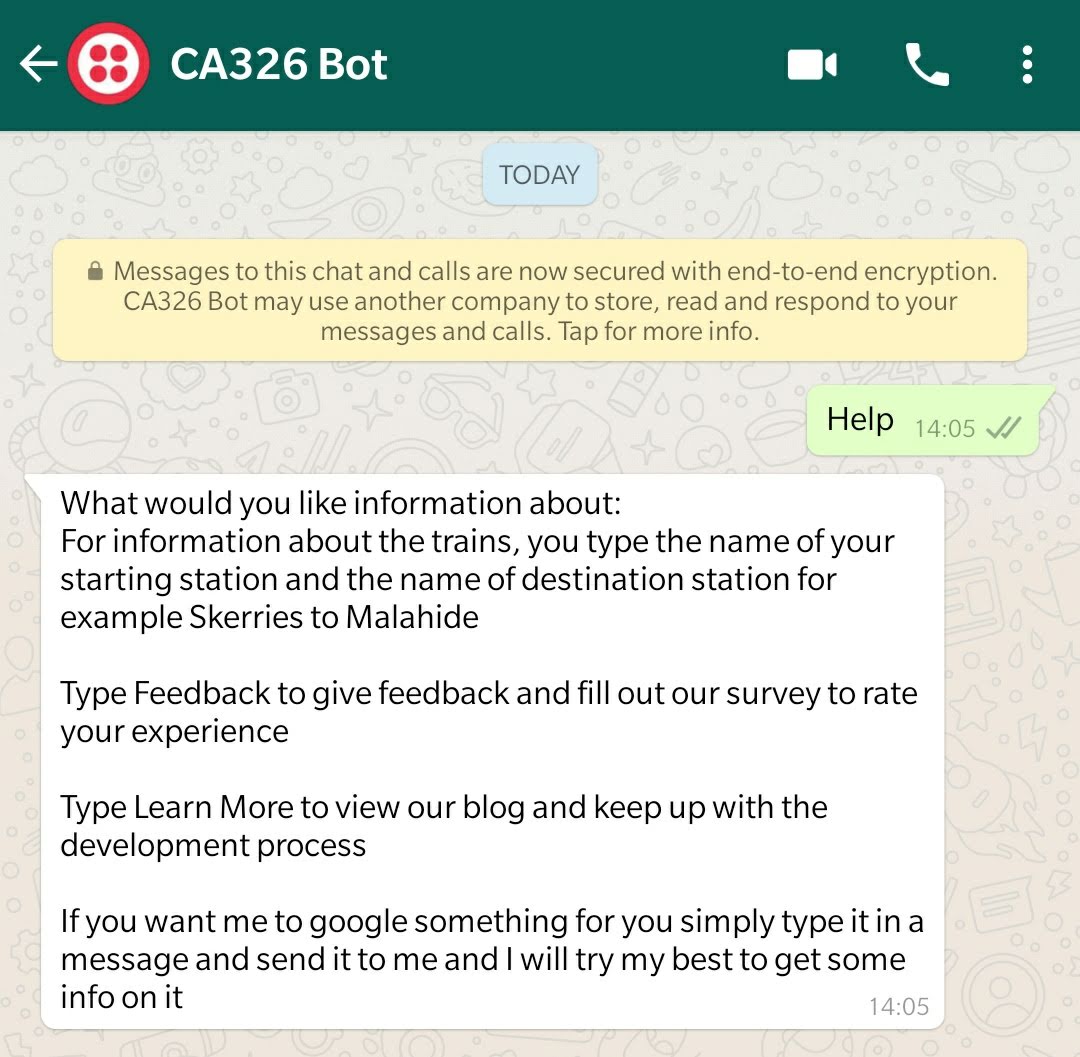
Once the chatbot is then set up the user can begin to text it, with whatever they wish to say. However, like most conversations the start will be a typical greeting scenario, so if the user were to send one of the numbers of greeting phrases we have prepared then it will trigger the bot to respond in a similar manner.



As you can see from the above screenshot the initial response to a small talk interaction is a hello and a brief description of what the tasks and functions the bot can carry out. It also lets the user know that if they were to type help it will return a list of functions they can use.

# Section 3 – The help function

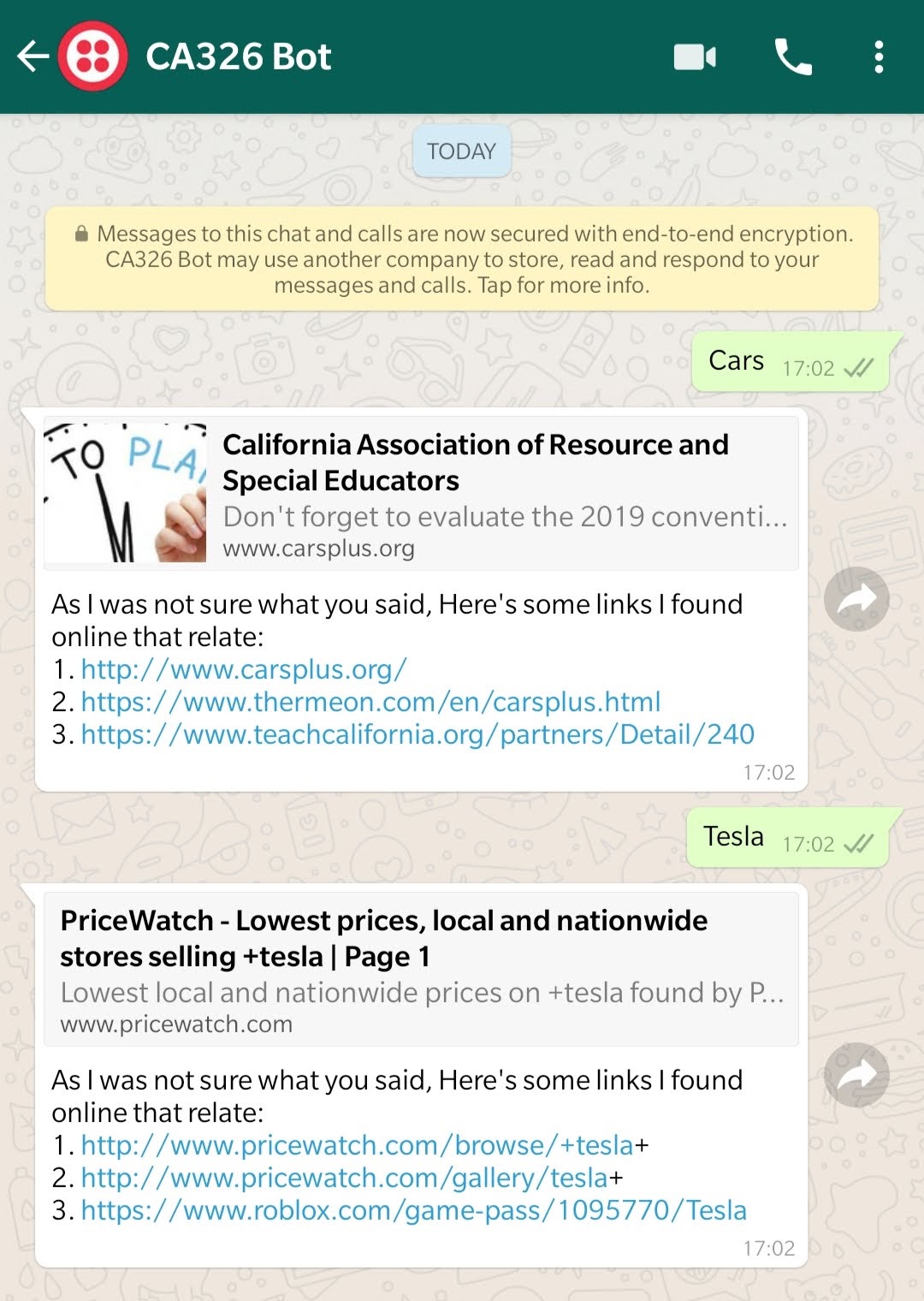
3.1 – Getting Help



As you can from the above screen shot if you simply send the bot a message containing the word “help”. It will return a message containing a list of the functions it can perform. It can give information on the trains if the user type in the start station and the destination. It will send a list of trains within the next 30 minutes.

If the user types “Feedback” it will send a link a survey to provide feedback on the service and recommend any improvements If the user types “Learn More” and sends it, it will respond with link to view the blog related to the development

# Section 4 – Google

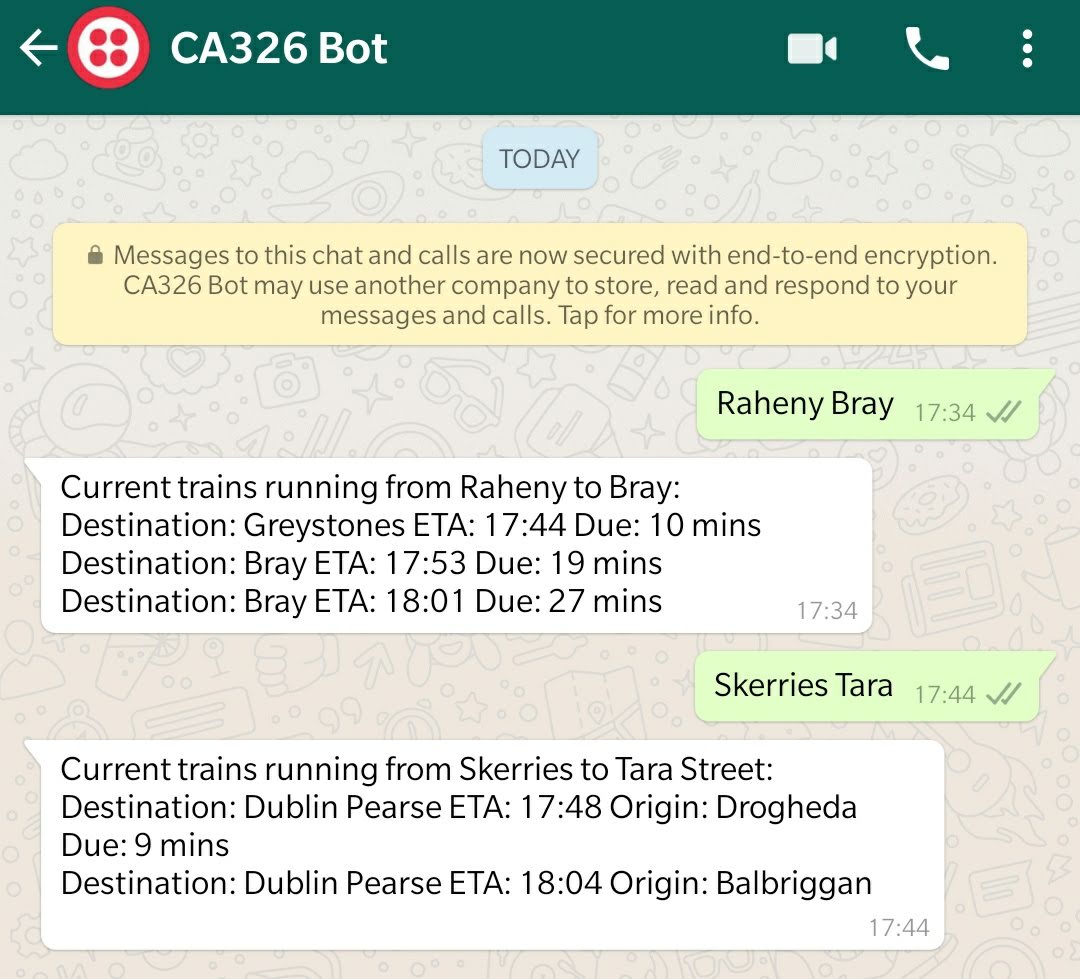
4.1 – Google within the chatbot

If the user wishes to have the chatbot do a basic Google search for a link to something such as “cars” all the user has to do is send a message containing the subject, they wish to search for and then the bot will query the Google search function in our back end and then return a few links related to the topic.

# Section 5 – Querying the Chatbot

5.1 – How to get information about the trains

The main function of our chatbot service is the querying of the Irish rail real time API. This done is by the user inputting the start and finish destination and then Dialogflow will recognise the entities and then passes them as parameters to an action on Dialogflow and our back end will then look for that specific action and then query the API.



As you can see in the above screen shot the user types in two stations and then the bot returns the trains that are due to go from Raheny to Bray in the next 30 minutes, also giving the Destination, the estimated time of arrival and how many minutes the train is due in.