10/4/2017

Done by: Tan Bing Xing Bryan U1721130G

Ang Hao Jie U1721621G

Wong Chong Xan, Alexander U1721864K

Trifena Caroline U1720618J

Darren Chua Qi Yang U1721600J

School: School of Computer Science and Engineering

CZ1003 Assignment

Telegram Bot Proposal

Introducing LinguiBot

Table of Contents

**INTRODUCTION**1

Background2

Problem Statement3

**PROPOSED SOLUTION**4

The idea5

Flowcharts6

Explanation7

Bot Manual8

**CONSTRAINTS AND LIMITATIONS**9

Point 1.10

Point 211

Point 312

Point 413

Improvements 14

**CONCLUSION**15

**Appendix**16

**Reference**17

1. **INTRODUCTION**
   1. **Background**

In today's world, globalisation is prevalent. We are often put in situations where we are forced to talk to people from other countries who may not understand our native language. This may lead to a language barrier during conversations. Language is always needed for any form of communication. It becomes a problem when people do not understand each other.

* 1. **Problem Statement**

Imagine conversing with a foreign friend or a business partner who doesn’t know English on Telegram. Since our native languages are different, we may not be able to express our points clearly in each other languages. Therefore, both parties would have difficulties in expressing themselves. A difference in language is the most common communication barrier that people tend to face, which can cause misunderstandings and misinterpretations in conversations.

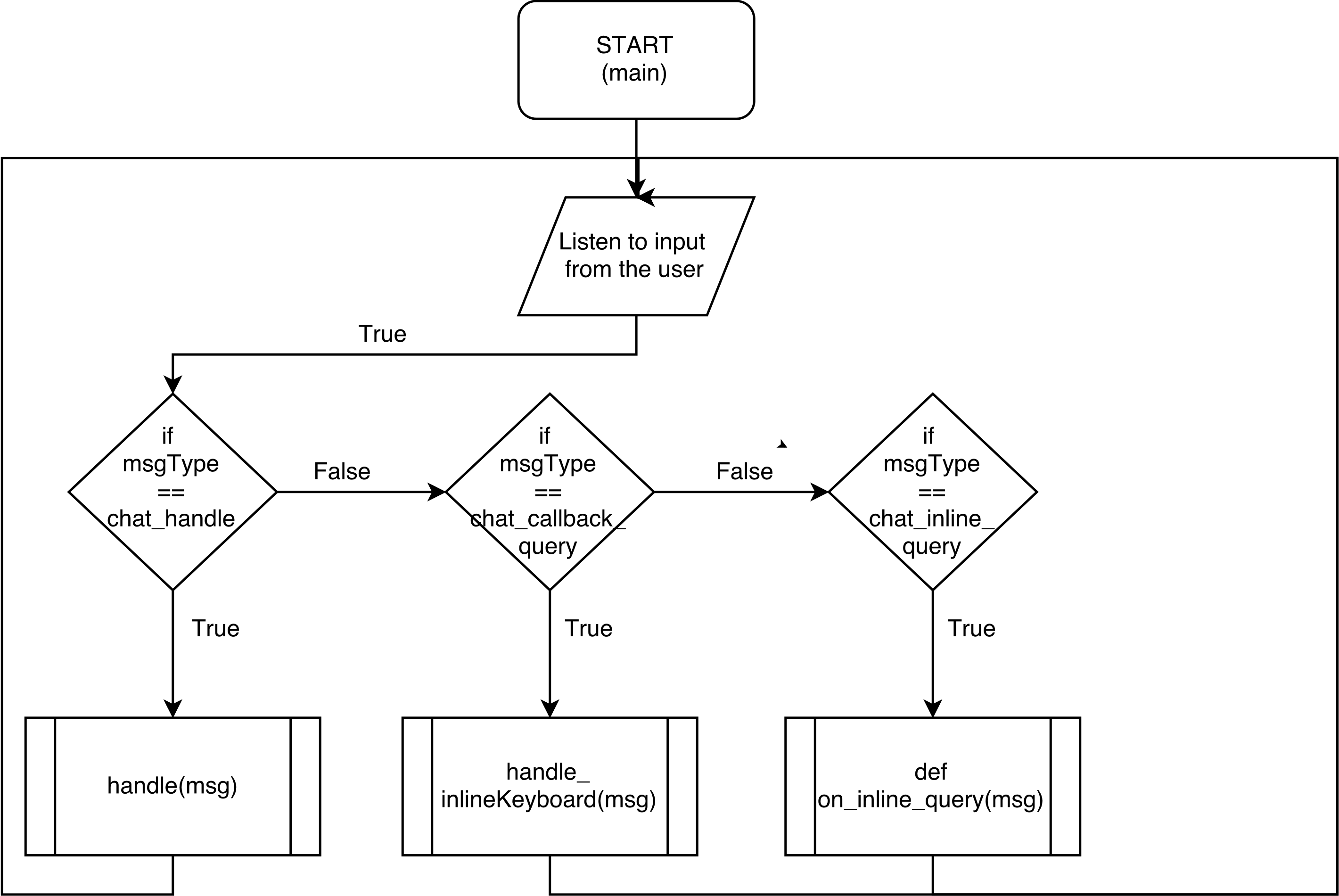
1. **PROPOSED SOLUTION**
   1. **The Idea**

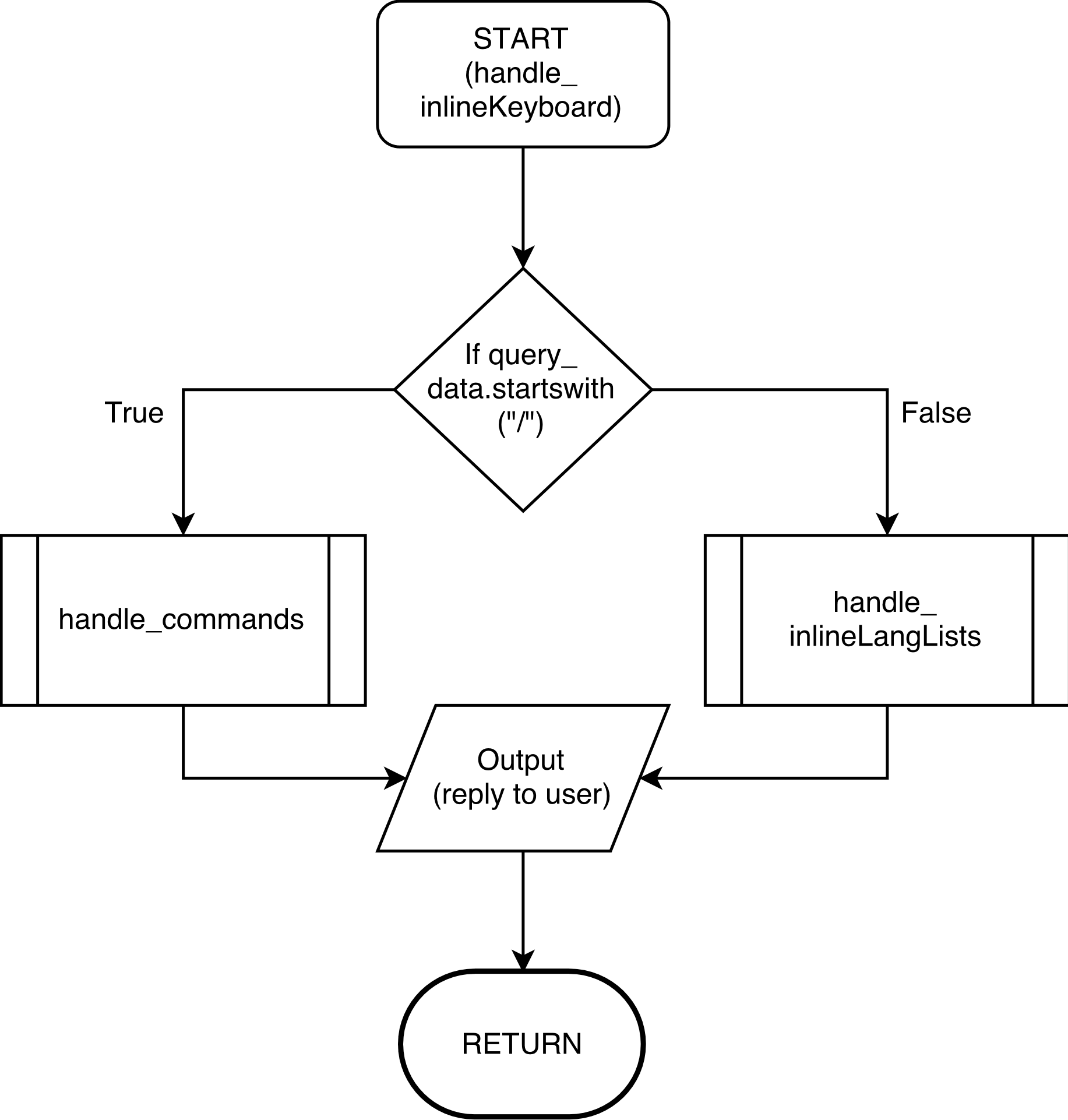
The proposed idea is the implementation of an in-app translator bot that is able to translate one language to other. With this solution, people can conveniently express their points in their own language, and have the other party understand your points clearly.

Introducing Linguibot, from the word, Linguist. It is a telegram bot that serves as a convenient translator and can translate any language to another language in conversations.

* 1. **Flowcharts**

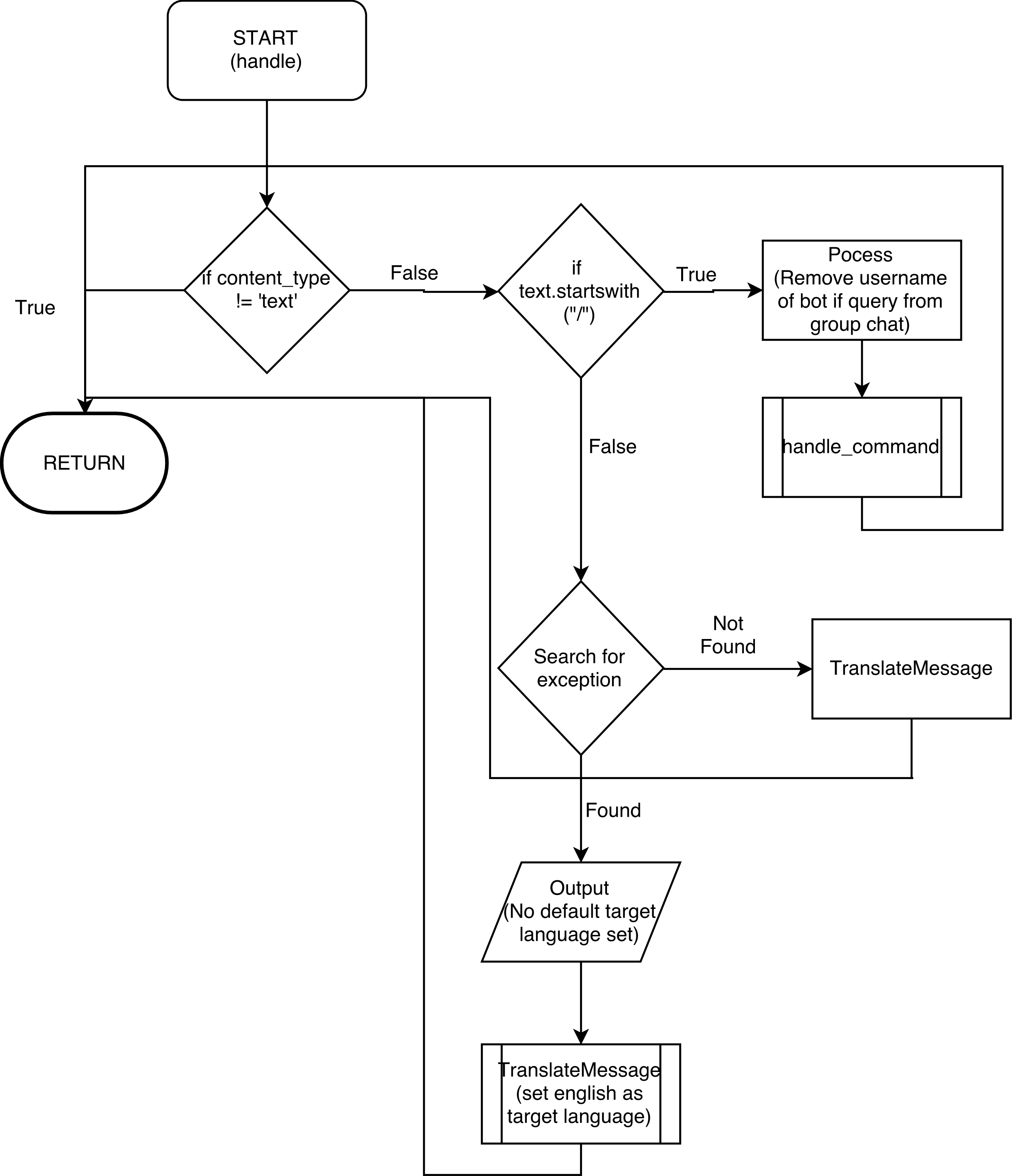
Main Program





**Handle inline keyboard flowchart**

**On\_inline\_query function flowchart**

**Handle flowchart**

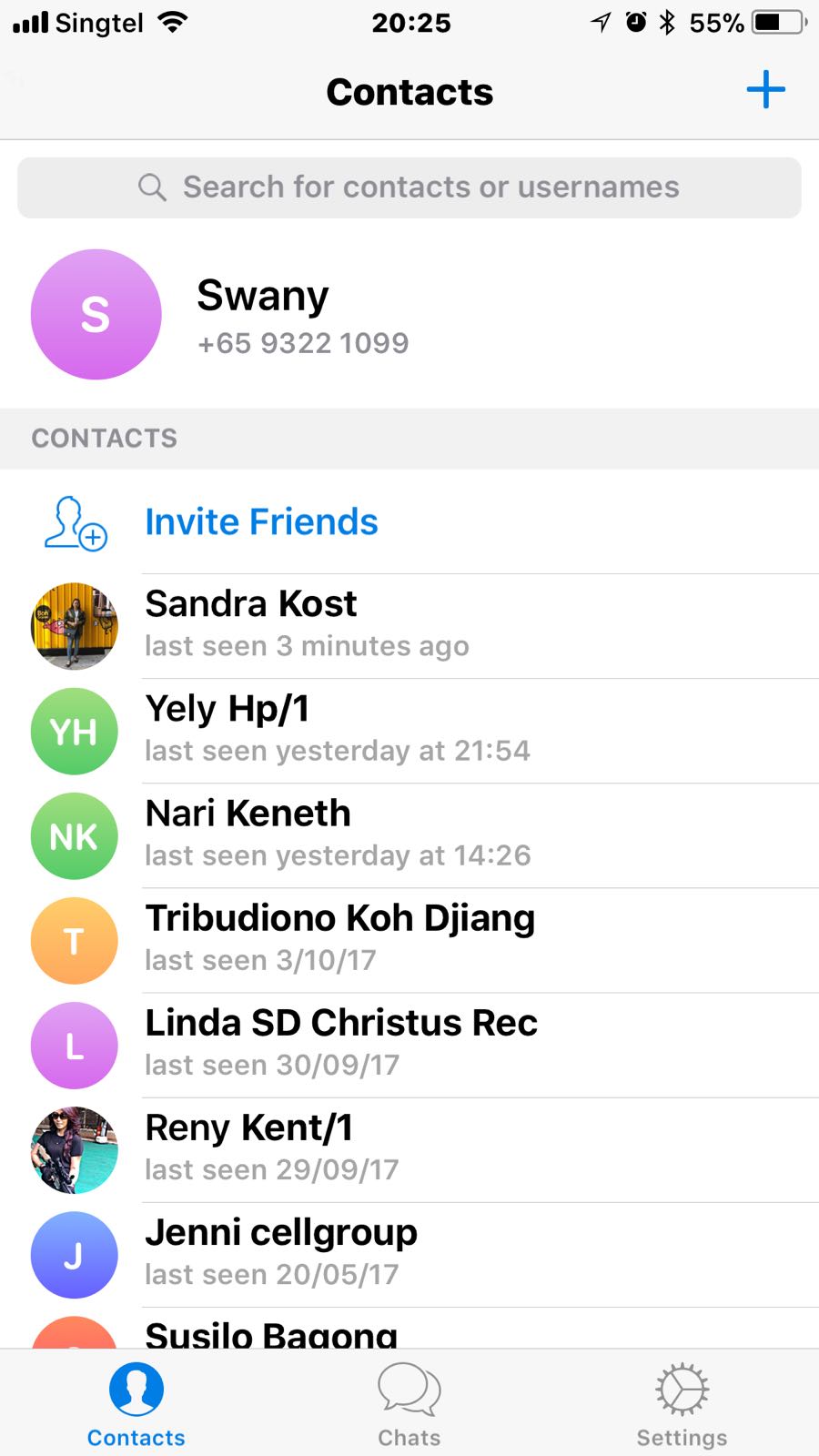
* 1. **Various Key Functions Explanation**

1. def handle(msg) **-** After receiving incoming message from clients (user who uses our bot), it will either call handle(msg) function or handle\_inlineKeyboard(msg) function. If it is normal message keyed-in using user’s own keyboard, this handle(msg) function will be called instead.
2. def handle\_commands(chat\_id, commands, msgType) **-** This function handles all the commands sent by the user, be it through normal chat or from inline buttons (reusability).
3. def handle\_inlineLangLists is the function that store all the starting alphabet of the translation language. This function will be call when the user presses the inline button of the respective alphabet. It is express in the statement block in the flowchart.
4. def handle\_inlineKeyboard(msg) **-** This function is called when the incoming data comes from inline keyboard buttons.
5. def on\_inline\_query(msg) is the function that handles all the query from the inline button on the application.
6. def handle(msg) is the function that handles all the normal chat.
7. def TranslateMessage(chat\_id, target, text) is the command that will translate all the messages detected to targeted language that the user choose.
   1. **Bot Manual**

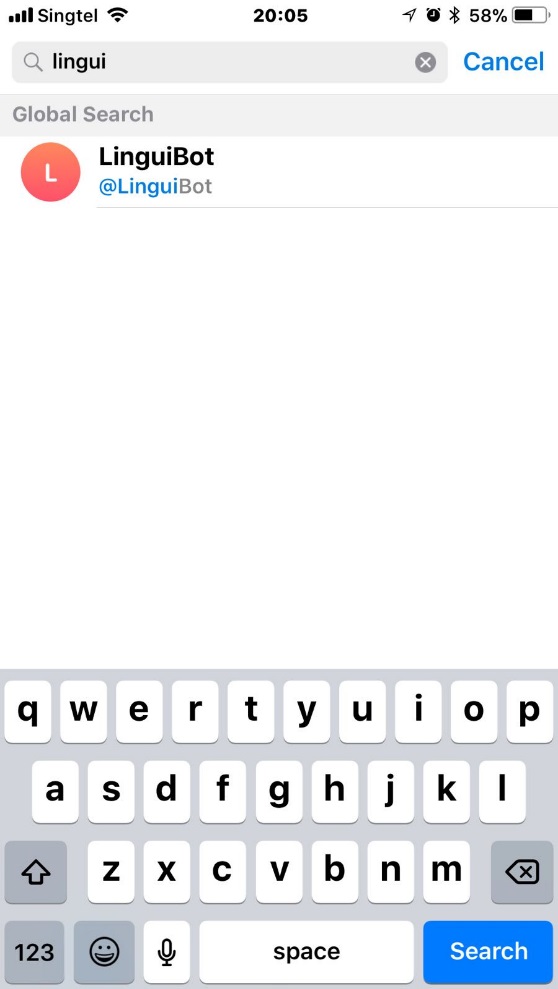
**How to setup?**



1. Open Telegram App
2. Go to the **Contacts** Tab



1. Search for “LinguiBot” in the search bar

****

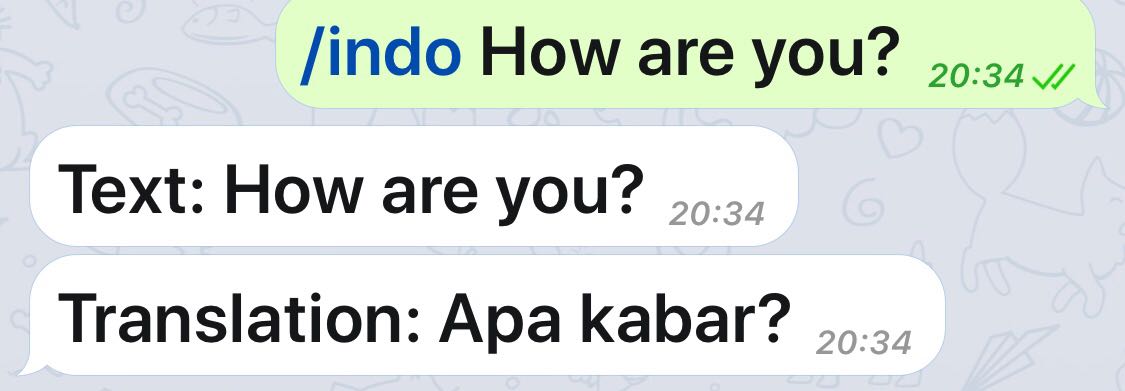
1. LinguiBot is ready to use!



**How to use?**

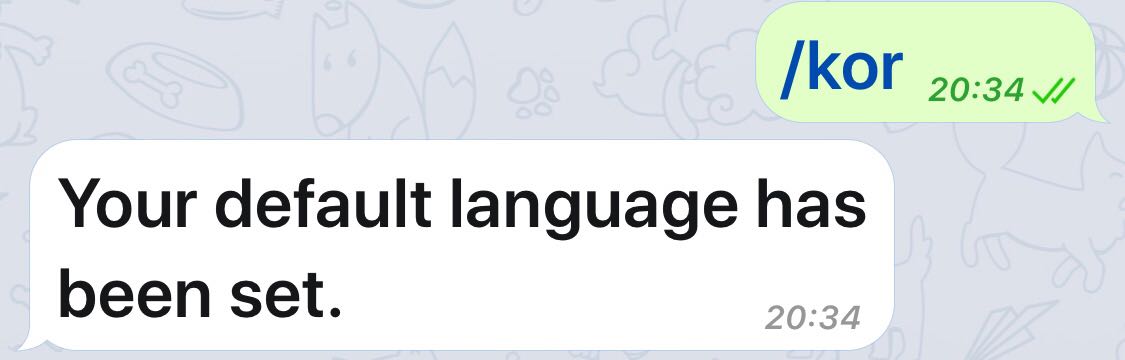
**QUICK TRANSLATE**

Type /(choice of language) (English text to be translated)

****For example, /indo How are you?

**LIST OF LANGUAGES**

Type /langlists

**SETTING A DEFAULT LANGUAGE**

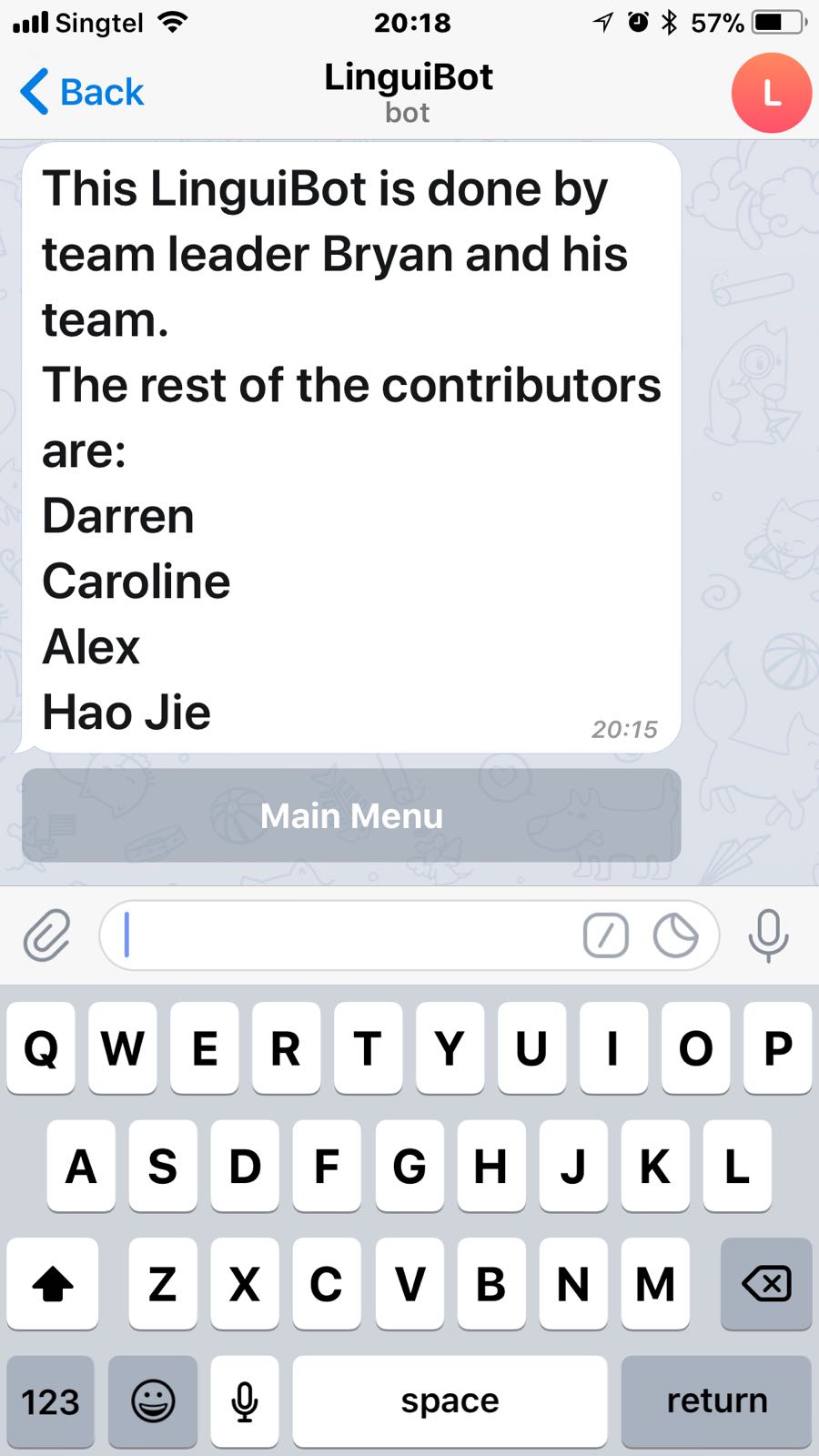
Method 1

Type /(choice of language)

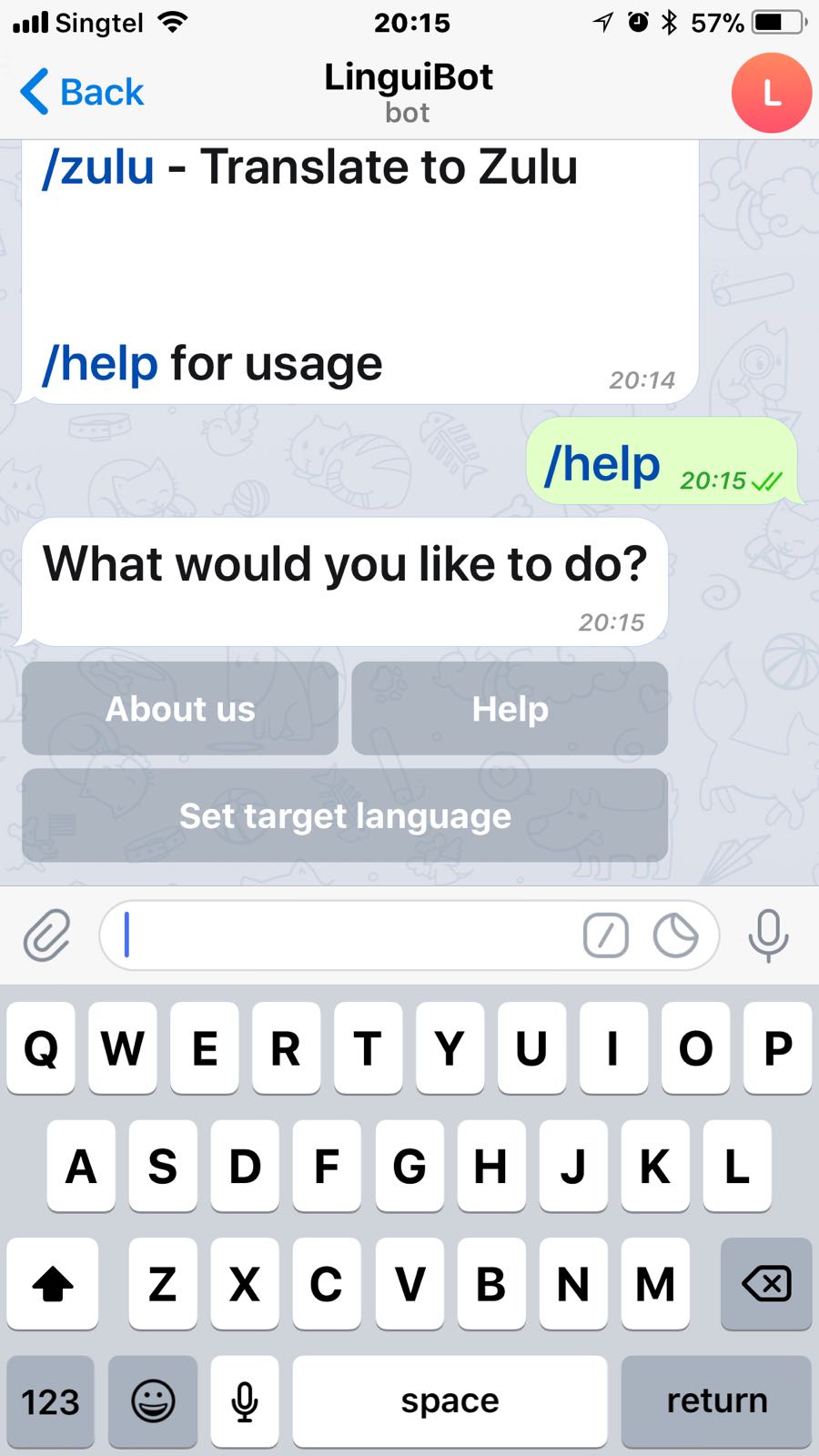
For example, /kor or /jap

Method 2

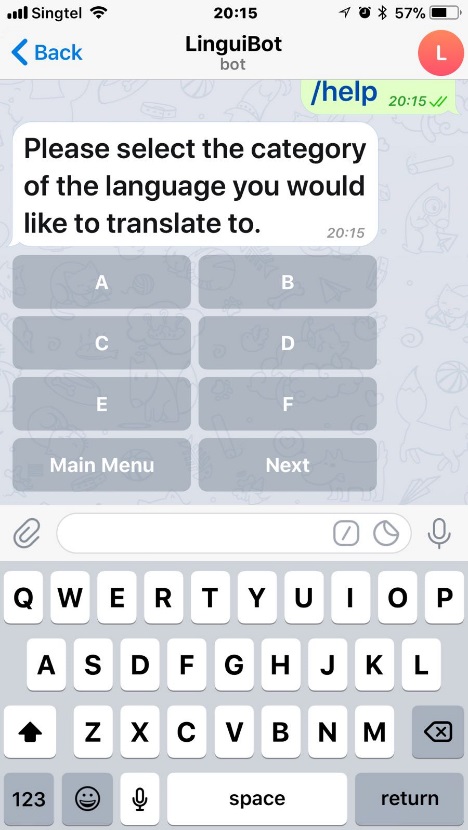
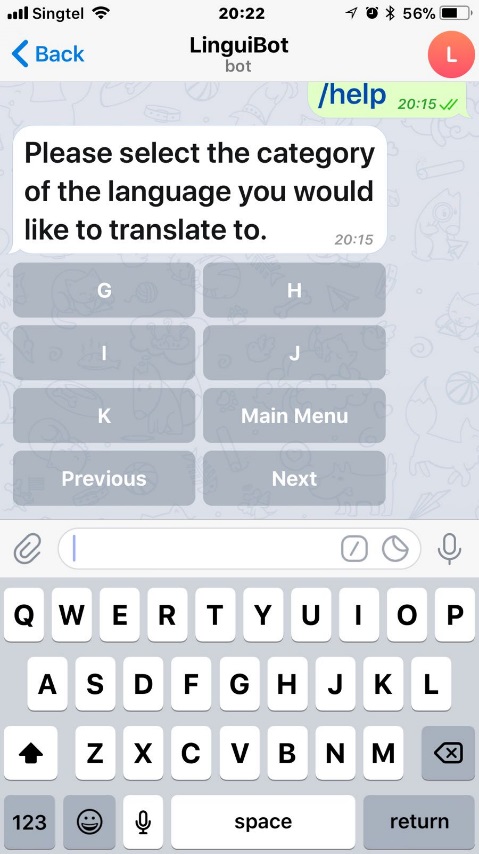
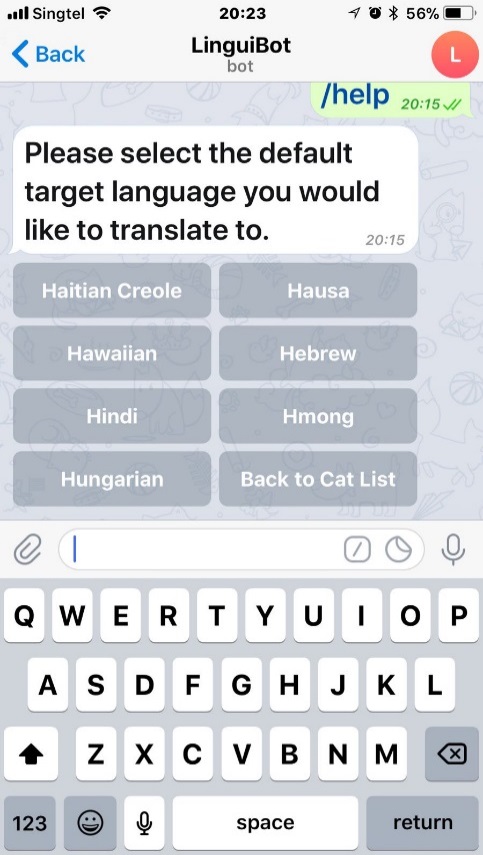
Go to Main Menu, if you are currently not in it.



Select **Set target language**



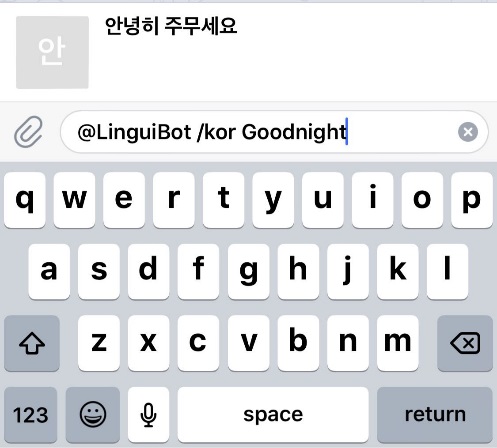
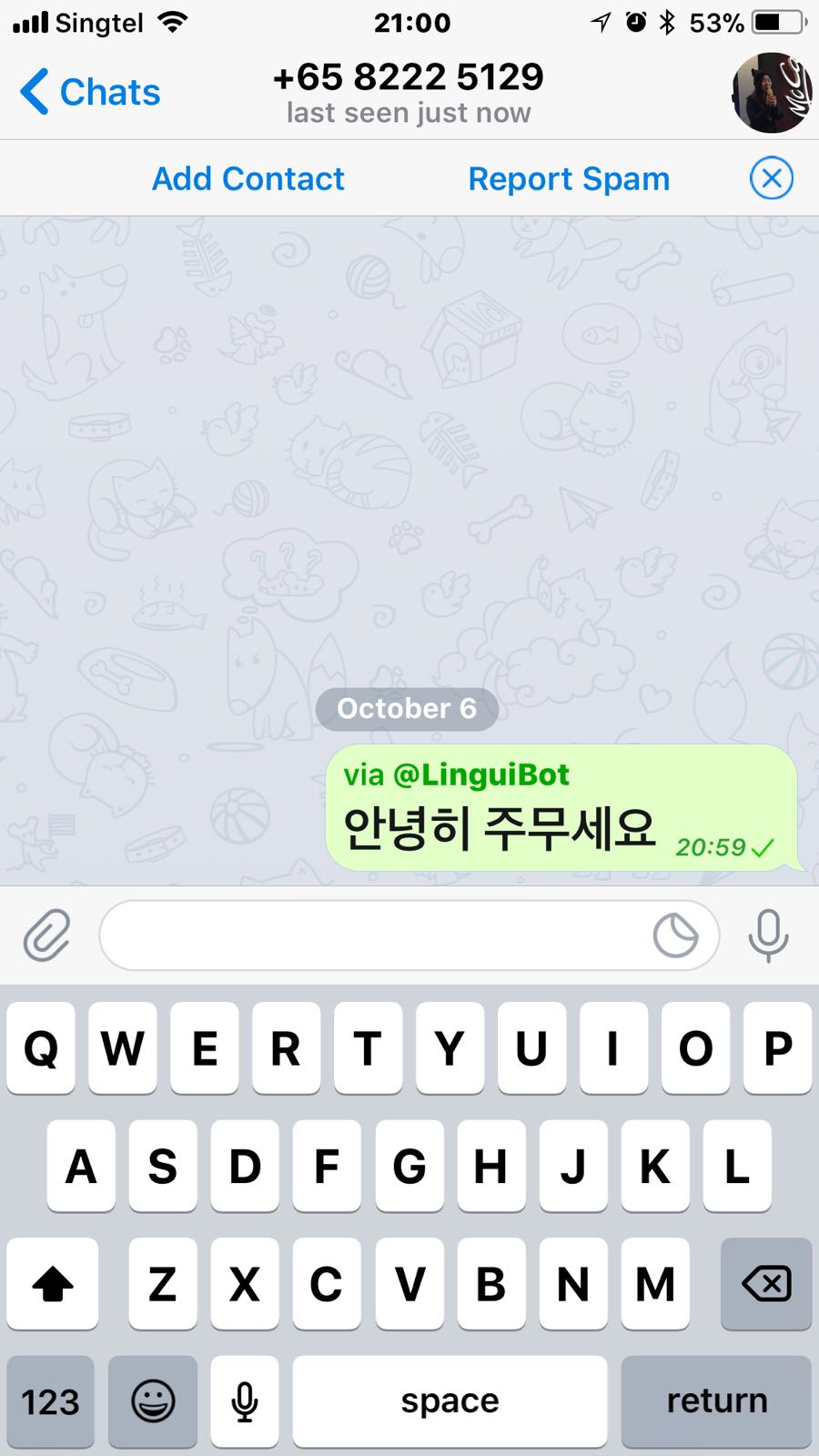
Select alphabet that chosen language starts with. For example, if I would like to choose Hungarian, select **Next** to find option **H** and pick **Hungarian.**

 ** **

*Select* ***Next*** *Select* ***H*** *Select* ***Hungarian***

**USING LINGUIBOT IN A PRIVATE CHAT**

Type @LinguiBot /(choice of language) (English text to be translated)

**USING LINGUIBOT IN A GROUP CHAT**

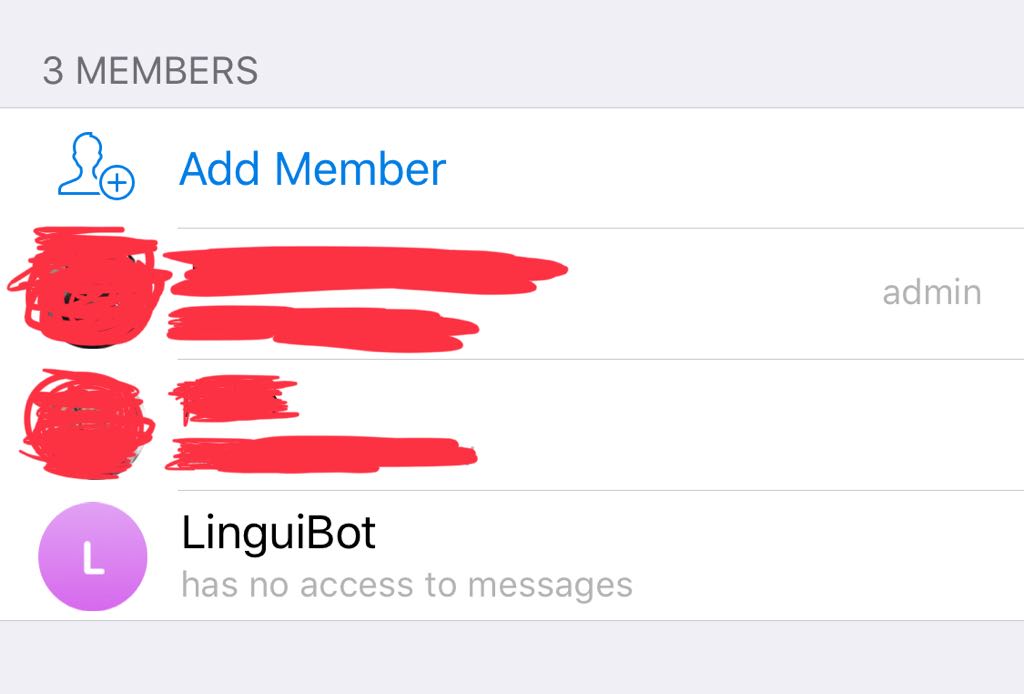
Method 1

Similar like using LiguiBot in a private chat,

type @LinguiBot /(choice of language) (English text to be translated)

Method 2

Add LinguiBot as a member in your group chat

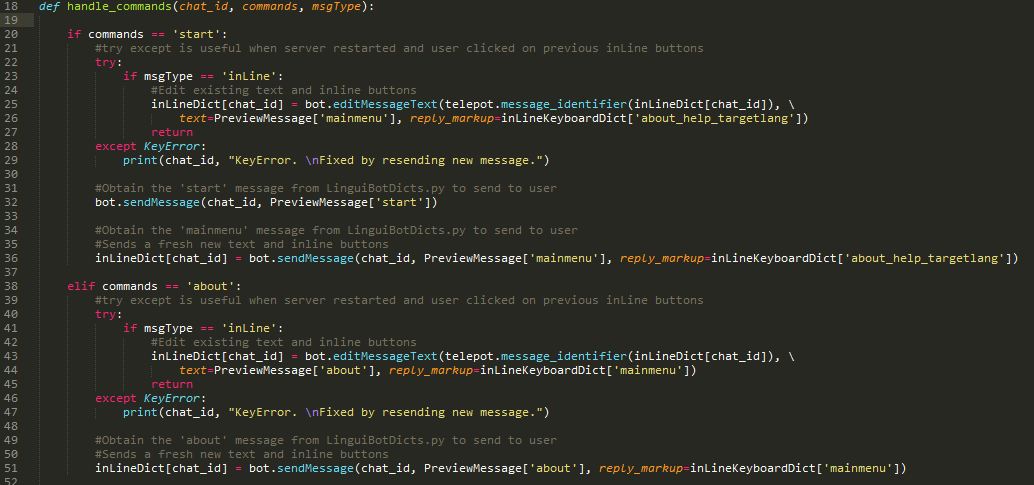
****

Use LinguiBot like doing quick translate by typing /(choice of language) (English text to be translated)

1. **CONSTRAINTS AND LIMITATIONS**
2. When building the bot, there was not enough detailed documentation. In Telepot, when a new function is added into the program, it does not give the variable types for the parameters, and it does not provide the return type of the function. There is a lack of explanation, and no examples were given. Thus, a huge amount of time was spent trying to figure out how to use the functions.

To tackle this issue, the functions and parameters were print out in ‘IDLE’ to find out what does each code do. We also manage to dig up a brief documentation with regards to some of the functions which helps a lot in explaining how does certain functions work.

1. It is not possible to obtain the other party's message in private chat. This results in both sides having to PM the bot for translation.
2. It is not possible to ensure reusability. Despite having the same lines of code, the output inline button and message sends to the user varies. Hence, it is tough to provide reusable of code in this function.



1. There are certain types of languages that have problems translating using the program. For example, from Malay to Spanish, there seems to be an issue with regards to translating from one language to the other while, from English to Spanish, it is possible.

 ****

* 1. **Improvements**

Instead of just being a translator bot, with given enough resources and money, we would like to implement a ‘Language’ teaching option which can teach different languages to users of the bot. Users can also choose to determine the purpose of their learning. For example, users can choose if they are learning for ‘Business/Work’, ‘Preparation for Language Test’ or ‘Self Improvement’. Teaching of language will be tailored to best suit users’ intended purpose of learning. This way, such targeted teaching will allow users to better earn fulfilment in learning their new language of choice when material taught is largely applicable to their intended use.

1. **Conclusion**

In conclusion, although there are many translation applications and software already existing, we felt that having to swap to a separate application or software for every sentence during a conversation would be tedious and would interrupt the flow of the conversation. Thus, we decided a convenient bot for translations would be helpful in maintaining the flow of conversation, which is important when trying to communicate and share ideas with someone, and reduce the extra effort required to swap to external applications to translate, thus making conversation with people from different cultures in general less tedious. We encountered a couple of difficulties when developing the bot so it is not perfect but it is functional.

Overall, this project has taught us what can programming is capable of, and there are endless possibilities in developing ideas to tackle issues in the world

1. **Appendix**

****

Figure 1 (Padilla, 2017) – Picture used for Telegram Bot Profile Picture

# References

Padilla, G. (2017, September 12). Localizing the editor: changing language.