Milestone 4

Q1. How is your project architecture related to the theory taught in the lecture?

The whole Line Bot system framework is made up of three parts:

1) Line Client, the terminal interface used by the users. Usually it can be a LINE Channel;

2) Line Server, it is used to receive the receive an interactive request from the Line Client;

3) Webhook Server, it is where the business logic, pass back component plan and the Line Bot SDK works.

Fig.1

When Line Client receives a message from the user, it will pass it to Line Server who act like a translator. Then the server will send this request to Webhook Server, where we have deployed our code solutions already. After getting the results, the results will be passed back to client through the server again. Each passing represents a HTTPS protocol transmission (a HTTPS Request + a HTTPS Response).

Inside the Webhook Server, has deployed our code already. In order to enrich the query analysis ability and the decision making capability, some outside service are necessary. Such as Google Dialogflow, Microsoft Azure Bot Service…

The hardware architecture of Heroku

HTTP reverse proxy: With Nginx, this layer only handles HTTP- level processing.

HTTP Cache: For static content, use Varnish for caching.

Routing Mesh: Architecture components implemented in Erlang, routing and addressing, to improve availability and scalability.

Dyno Grid: The code deployed by the user runs here, which can be simply regarded as an application server cluster environment, but with a smaller granularity.

Database: PostgreSQL can also use a remote database.

Memory Cache: Memcached is a necessity for home travel architecture.

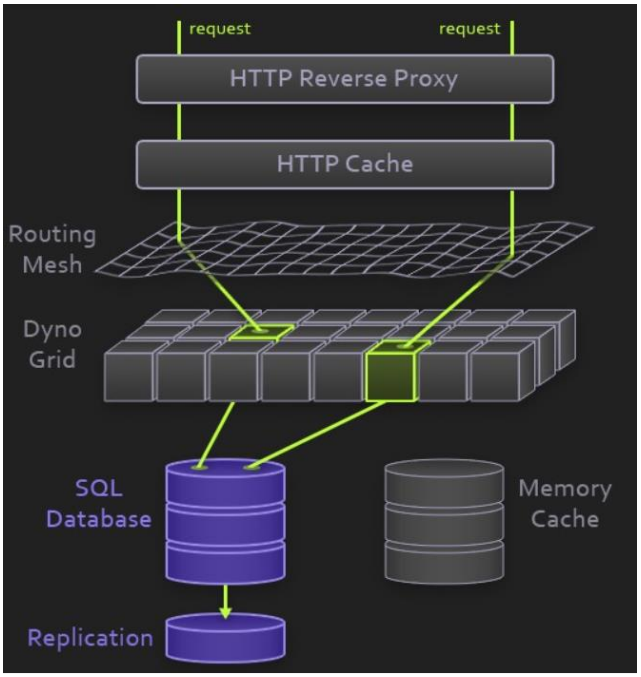


Fig.2

Python code framework Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program.

The simple use demo of Flask:

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

Fig.3

Q2. Can you demonstrate, with some screen cap, how to increase capacity of your chat bot service?

Method 1. Extend the access frequency in Messaging API.

To prevent massive access, each LINE Messaging API account has different API access frequency limits depending on the scheme, as shown in the following table:

方案	呼叫頻率限制	收訊者數量頻率限制
Developer Trial	1,000/min	20,000/min
其他方案	10,000/min	200,000/min

Fig.4

When the access frequency of the Messager API exceed the threshold value, the API will response a「429 Too Many Requests」fault. If that happen, we can change the scheme in the API.

Here is the statistic for the access frequency:

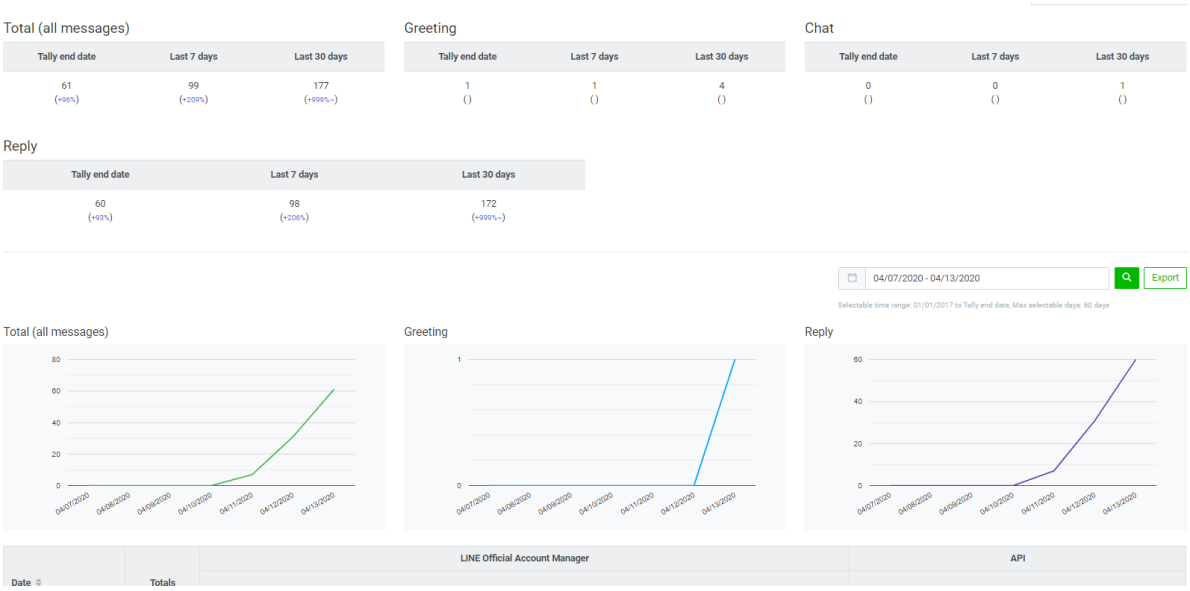


Fig.5

Method 2. Scale out/up in Heroku.

We can also increase our need by changing the scheme in Heroku to increase the capability.

a. Scale up on each server. (Vertical scaling)

	Free	Hobby	Standard 1x	Standard 2x	Performance M	Performance L
	$0	$7/dyno per month	$25/dyno per month	$50/dyno per month	$250/dyno per month	$500/dyno per month
			PROFESSIONAL			
What is it good for?	Ideal for experimenting with cloud applications in a limited sandbox.	Perfect for small scale personal projects and hobby apps.	Enhanced visibility, performance, and availability for powering your production applications.		Superior performance when it's most critical for your super scale, high traffic apps.	
RAM	512MB	512MB	512MB	1GB	2.5GB	14GB

Fig.6

b. Scale out by adding more servers (Horizontal scaling)

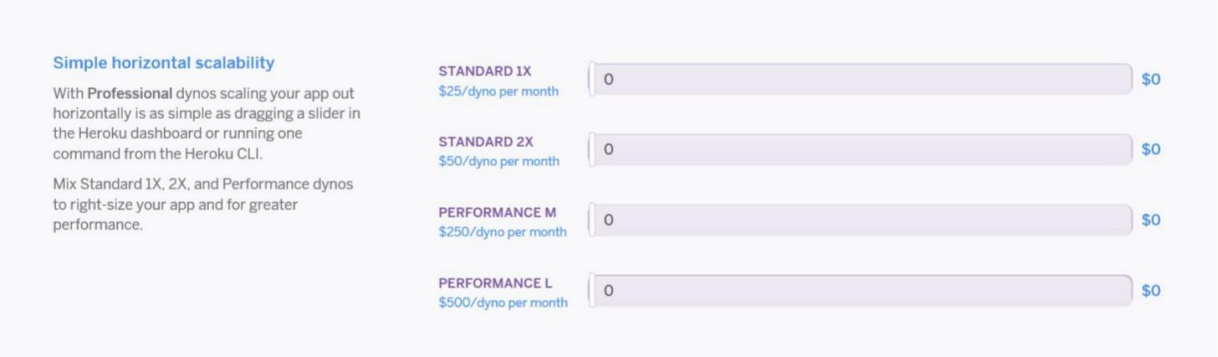


Fig.7

Method 3. Rewrite the python code to get better operating performance.

a. Write the python code with smaller time complexity and space complexity

Q3. Can you identify if your bot is one of the examples of PaaS, IaaS, SaaS? Explain your answer.

What are IaaS, PaaS and SaaS?

IaaS: cloud-based services, pay-as-you-go for services such as storage, networking and virtualization.

PaaS: hardware and software tools available over the internet

SaaS: software that's available via a third-party over the internet.

My bot is based on a PaaS. PaaS is often used when a developer wants to create a unique application in a most cost-effective and time-effective way. I used Heroku to deploy my app. Heroku itself is a cloud platform as a service (PaaS) supporting several programming languages including python, which can be considered as a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages.

Several Advantages by using a PaaS platform:

- Accessible by multiple users. – Distributed system and router mechanism.
- Scalable – I can choose from various tiers of resources to suit the size of my business.
- Built on virtualization technology. – Better understanding and control of the physical resources.
- Easy to run without extensive system administration knowledge. – I can focus my time in creating, testing, and deploying the app instead of infrastructure management

I also use some SaaSs to enrich my bot function. The Messager API, Amap API I used can be considered as the SaaS, since they provide a complete function with API which I can call. And it is no need for me to figure out the implementation details.

The Redis of the Heroku add-on is a Database-as-a-Service. It provides in-memory data structure store, which enables data persistence and high availability through replication and backups. Supporting various data structures such as strings, hashes, lists ...

Q4. Present your work on stage for not more than 5 minutes

See in the video.

5. Screenshot of all the functions of bot Firegod

Response 1：CarouselTemplate Response for key word “Hello”

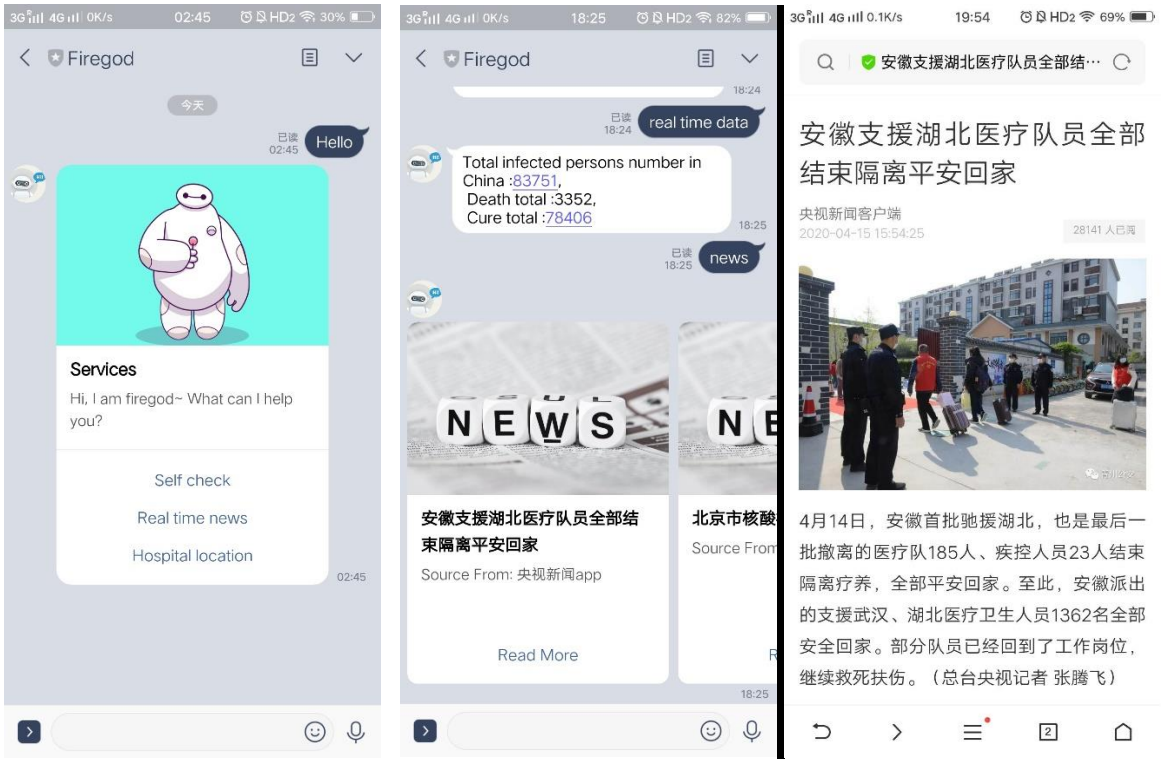
It will show the three main function in our bot, self check, real time new, and nearest hospital.

Response 2: Text Response of key word “real time data”

If I type “real time data”, the bot will return us a text about the number of total infected, total death and total cure people in real time that get from the internet through the json API.

Response 3: Text Response of key word “real time news”

If I chick the real time news button or typing “real time news”, the bot will automatically send the “news” text for me. Then the Firegod will return the three most recent news from the internet in the form of CarouselTemplate, which can be slipped horizontally and the link of “Read More” can take you to the URL of the original resources.



Response 4: Location Response of key word “nearest hospital to <location_name>”

If I chick the real time Hospital location button or typing “nearest hospital to”, the bot will return the location of the nearest hospital to Hong Kong Baptist University by default, emphasizing the location with a red marker on the map. If I type the key word of “nearest hospital to 北京理工大学” for example, the bot will return the nearest hospital to the Beijing Institute of Technology.

This function using the Searching POI function that provided by the AMap API, which is one of the outside server rather than Redis.

Response 5: Using Redis to remember the username.

As you can in screenshot. At the beginning, when user say “Hi” to Firegod, Firegod will respon “Hi, what can I help you”. At this time, if we type “get name” to the bot, the bot will reply “You haven’t set name, please try add name:<YOUR_NAME> first.” Then the user will accordingly send the text of add name:<tianxin> for an example. Then the bot will response “You have set your name as: tianxin”. After which, when I say “Hi” again to Firegod, Firegod will remember my name and say “Hi, tianxin~ what can I help you?” (<tianxin> is the name that have been store in the Redis database)

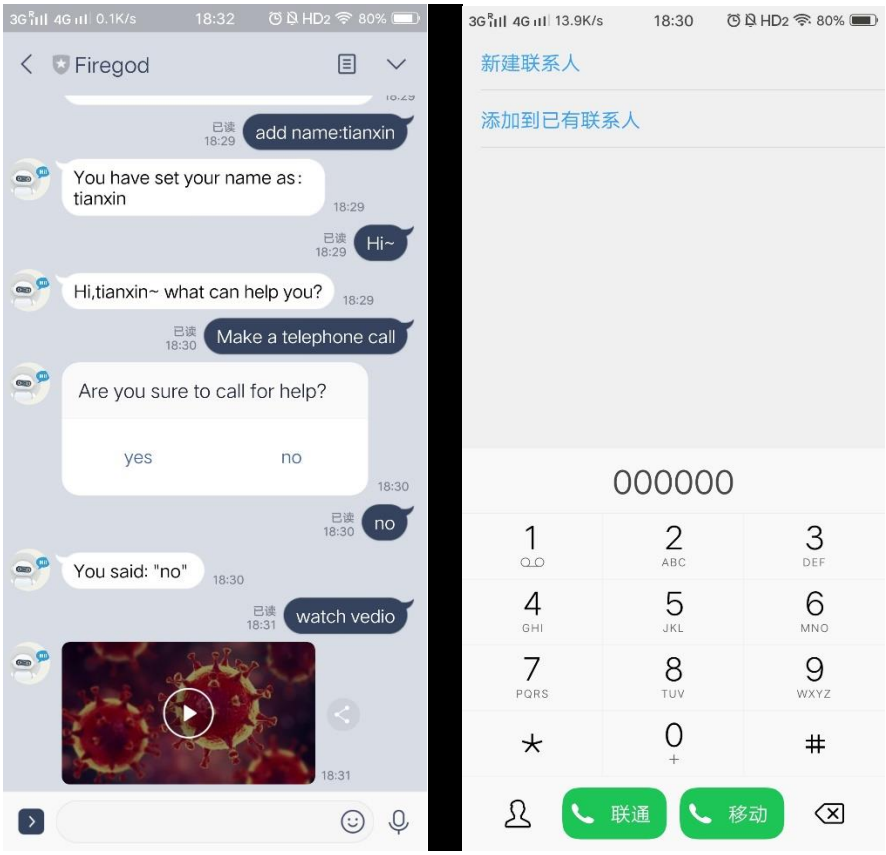
In this function, Redis is used to store the permanent data according to the user_id that get from the LINE Messager API.

Response 6: Call local department of health to get help

During the 14-days quarantine, people may need help of the local department of health. Then they can ask the bot for help to make a telephone call to the department of health.

Response 7: Watch the popular science film of COVID-19

This function will response to the key word “video”. Then the bot will send a video message to user.



Response 8: Query types of Image, video, Sticker will get a response of “Nice image/video/Sticker” responsively.

Response 9: Other text queries will get an echo response

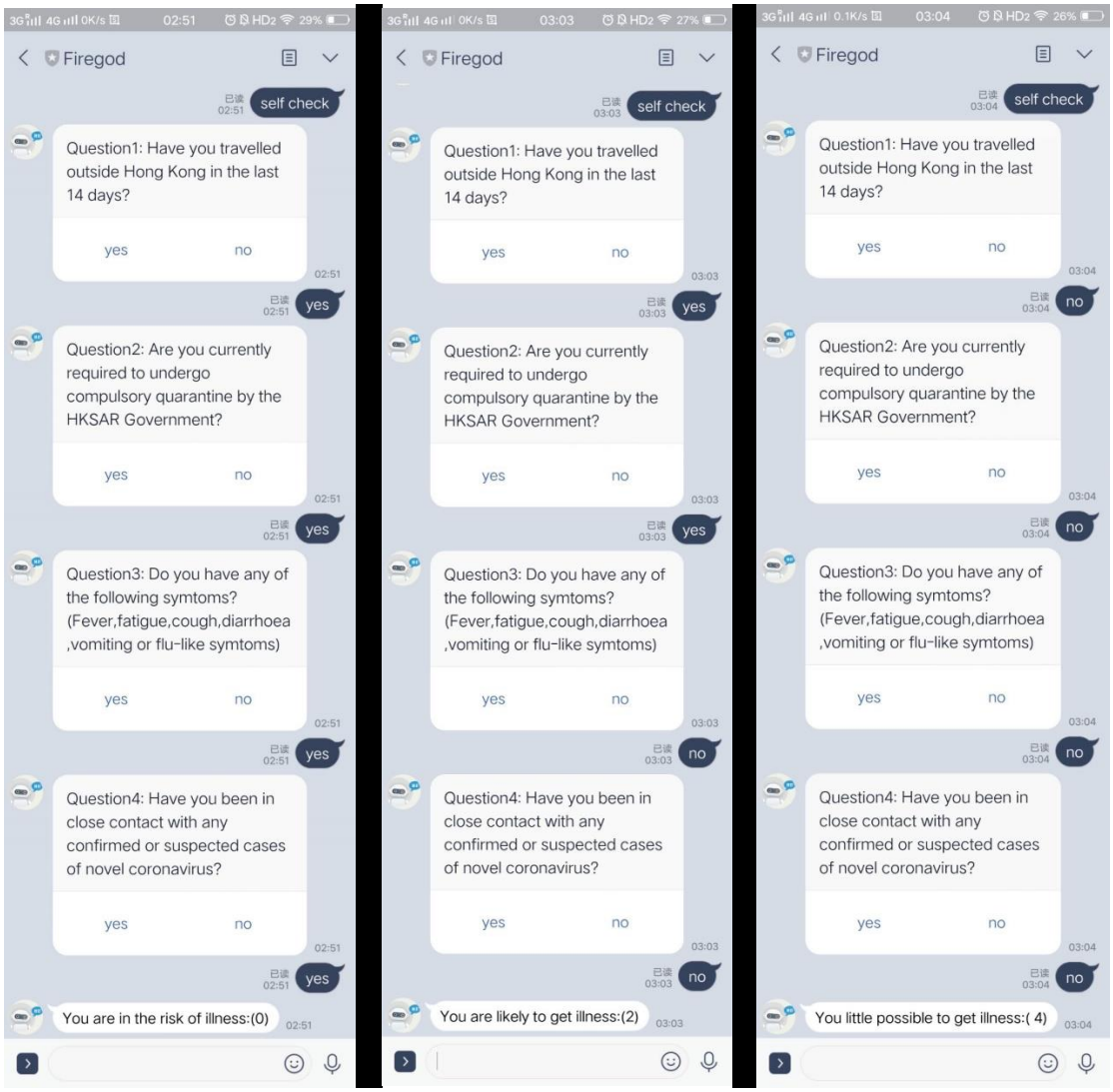
Response 10: Self-checking function (The use of postbackEvent)

If I chick the Self check button or typing “self check”, the bot will response me a series of questions about the user’ s information to make a judgment how high the risk of this user will get illness. There are totally four questions. Every time user gives a “no” answer, the bot will mark +1 at backend. Then finally get the judgment according the mark amount.

If mark is 4, then the bot will return “You little possible to get illness”

If 1<mark <4, then the bot will return “You are likely to get illness”

If mark<=1, then the bot will return “You are in the risk of illness:”


Firegod
self check
Question1: Have you travelled outside Hong Kong in the last 14 days?
yes
no
yes
Question2: Are you currently required to undergo compulsory quarantine by the HKSAR Government?
yes
no
yes
Question3: Do you have any of the following symtoms? (Fever,fatigue,cough,diarrhoea ,vomiting or flu-like symtoms)
yes
no
yes
Question4: Have you been in close contact with any confirmed or suspected cases of novel coronavirus?
yes
no
yes
You are in the risk of illness:(0)
Firegod
self check
Question1: Have you travelled outside Hong Kong in the last 14 days?
yes
no
yes
Question2: Are you currently required to undergo compulsory quarantine by the HKSAR Government?
yes
no
yes
Question3: Do you have any of the following symtoms? (Fever,fatigue,cough,diarrhoea ,vomiting or flu-like symtoms)
yes
no
no
Question4: Have you been in close contact with any confirmed or suspected cases of novel coronavirus?
yes
no
no
You are likely to get illness:(2)
Firegod
self check
Question1: Have you travelled outside Hong Kong in the last 14 days?
yes
no
no
Question2: Are you currently required to undergo compulsory quarantine by the HKSAR Government?
yes
no
no
Question3: Do you have any of the following symtoms? (Fever,fatigue,cough,diarrhoea ,vomiting or flu-like symtoms)
yes
no
no
Question4: Have you been in close contact with any confirmed or suspected cases of novel coronavirus?
yes
no
no
You little possible to get illness:(4)

References

[1] Document of Heroku Dev Centre; Available from https://devcenter.heroku.com/articles/how-heroku-works

[2] Blog of Fenng, Learning Heroku Structure; Available from <https://dbanotes.net/arch/heroku_architecture.html>

[3] Document of LINE Developer; Available from https://developers.line.biz/en/docs/messaging-api/overview/#how-it-works

[4] Tony Hou, IaaS vs PaaS vs SaaS Enter the Ecommerce Vernacular: What You Need to Know, Examples & More; Available from <https://www.bigcommerce.com/blog/saas-vs-paas-vs-iaas/#the-key-differences-between-on-premise-saas-paas-iaas>