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

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. It is a system in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing.

We learned three different **Cloud Delivery Models** PaaS, IaaS, SaaS in this course. The platform of this project is Heroku, which is a kind of Platform as a service (**PaaS**). Heroku also provides the domain name for this project. The domain name server helps a client to resolve the IP address of a domain name. In order to enrich the functions of chatbots, we invocated the website API to provide us with more data. These services are considered as **SaaS**.

We implemented Redis to store information. **Redis** is a key-value **cloud storage system**, and it is also commonly referred to as a data structure server. Redis supports synchronizing data to multiple slave libraries, this feature is very beneficial to improve read performance. In addition, we deployed Alice chatbot in this project. When it provides services, it will save the information input by the user to the session. **Figure 1** shows the storage method of distributed session sharing. The storage method includes the Redis and Session, and the basic idea is to move the storage location of the session back from the server to a common third-party storage (such as Redis).

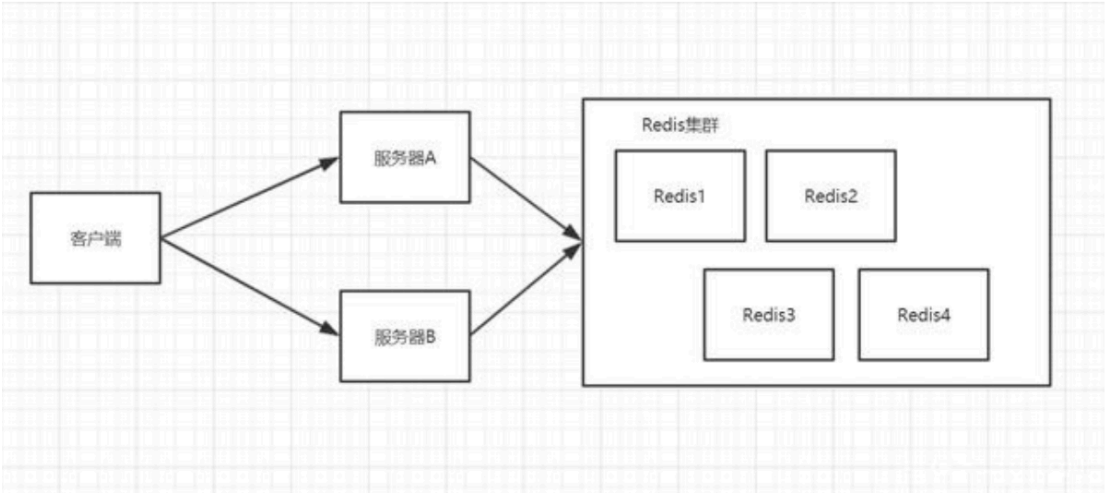

Figure 1. the storage method of distributed session sharing

Furthermore, we also learned serval **communication paradigms** in this course. Our project applied the Remote Invocation which is the most common one, based on the two-way exchange between communicating entities.

The final interface is provided by Line. Through the assignment 2, we have the experience to deploy the Python **Line chatbot** to Heroku platform.

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

In this project, considering practicality and user-friendliness, we first deployed the Alice chatbot to complete daily conversation. **Figure 2** and **Figure 3** show the part code of Alice chatbot and some simple questions and answers, such as greetings in the Line bot respectively.

```
print(os.path.abspath('../templates'))
app = Flask(__name__, template_folder=os.path.abspath('../templates'))

alice_path = get_module_dir('aiml') + '/botdata/alice'
# 切换到语料库所在工作目录
os.chdir(alice_path)
alice = aiml.Kernel()
alice.learn("startup.xml")
alice.respond('LOAD ALICE')
```

Figure 2. the part code of Alice chatbot

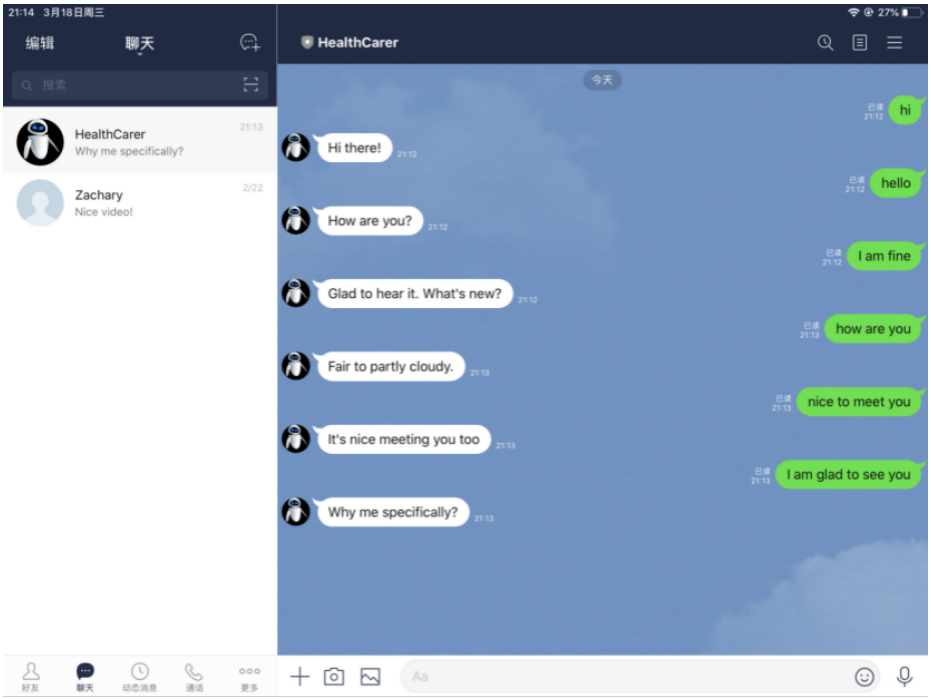

Figure 3. some simple questions and answers, such as greetings in Line bot

The next function is Autonavi map. The users may want to obtain the location information of COVID-19 patients. We crawl relevant data through websites such as the Hong Kong Department of Health, and then address information will be converted to latitude and longitude. The Autonavi generate maps based on latitude and longitude.

We first use the web services API, geographic/inverse geocoding, to convert location information to latitude and longitude. **Figure 4** describes the code of converting location information to latitude and longitude.

```
def geocode(location):
    parameters = {'address': location, 'key': '69b47d1b0be4c946ffbbd65e705b6320'}
    base = 'http://restapi.amap.com/v3/geocode/geo'
    response = requests.get(base, parameters)
    answer = response.json()
    if len(answer['geocodes']) == 0:
        return None
    return answer['geocodes'][0]['location']
```

Figure 4. the code of converting location information to latitude and longitude

The patient information crawled from the website will be stored in Redis. **Figure 5** shows the code of storing patient information in Redis.

```
def get_locations():
    url = 'https://www.chp.gov.hk/files/pdf/building_list_chi.pdf'
    r = requests.get(url)
    my_file = "building_list_chi.pdf"
    with open("building_list_chi.pdf", "wb") as code:
        code.write(r.content)

    pdf = pdfplumber.open("building_list_chi.pdf")
    locations = []
    for page in pdf.pages:
        # print(page.extract_text())
        for pdf_table in page.extract_tables():
            table = []
            cells = []
            for row in pdf_table:
                if not any(row):
                    # 如果一行全为空，则视为一条记录结束
                    if any(cells):
                        table.append(cells)
                        cells = []
                elif all(row):
                    # 如果一行全不为空，则本条为新行，上一条结束
                    if any(cells):
                        table.append(cells)
                        cells = []
                    table.append(row)
                else:
                    if len(cells) == 0:
                        cells = row
                    else:
                        for i in range(len(row)):
                            if row[i] is not None:
                                cells[i] = row[i] if cells[i] is None else cells[i] + row[i]
            for row in table:
                line = [re.sub('\s+', '', cell) if cell is not None else None for cell in row]
                locations.append("香港" + line[1])
                print(line)
            print('---------- 分割线 ----------')
```

Figure 5. the code of storing patient information in Redis

Figure 6 shows the code of constructing map.

```
map = '''

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="chrome=1">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <style type="text/css">
        body, html, #container {{
            height: 100%;
            margin: 0px;
            font: 12px Arial;
        }}

        .taiwan {{
            border: solid 1px red;
            color: red;
            float: left;
            width: 50px;
            background-color: rgba(255, 0, 0, 0.1)
        }}
    </style>
    <title>Confirmed COVID-19s in past 14 days</title>
</head>
<body>
<div id="container" tabindex="0"></div>
<script src="//webapi.amap.com/ui/1.0/main.js?v=1.0.11"></script>
<script src="https://webapi.amap.com/maps?v=1.4.15&key=999bd05545f336fce91acad15209da89"></script>
<script type="text/javascript">
```

Figure 6. the code of constructing map

Except for providing COVID-19 map for users, we also implement the news recommendation. **Figure 7** shows the code of recommending related news for users. **Figure 8** shows the screen cap of COVID-19 map function and news recommendation in Line bot.

```
def crawl_news(number=3):
    head = {
        "accept": "text/html, */*; q=0.01",
        "accept-encoding": "gzip, deflate, br",
        "accept-language": "en-HK,en;q=0.9,zh-HK;q=0.8,zh-CN;q=0.7,zh;q=0.6,en-US;q=0.5",
        'referer': 'https://www.news.gov.hk/chi/categories/covid19/index.html',
        'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.
    }
    params = {
        'language': 'chi',
        'category': 'covid19',
        'max': number,
        'returnCnt': '100',

    }

    page = 'https://www.news.gov.hk/jsp/customSortNewsArticle.jsp'

    s = requests.Session()
    s.get(page, headers=head)
    s.headers.update(head)
    r = s.get(page, params=params)
    # print(r)

    soup = BeautifulSoup(r.text, 'xml')
    items = soup.find_all('item')
    # print(r.text)
    news = []
    for item in items:
        title = item.find('title').text
        img_url = 'https://www.news.gov.hk' + item.find('landingPagePreviewImage').text
        summary = item.find('articleSummary').text
        article_url = 'https://www.news.gov.hk' + item.find('generateHtmlPath').text
        news.append([title, img_url, summary, article_url])
    return news
```

Figure 7. the code of COVID-19 news recommendation

Figure 8. the function of COVID-19 map and news recommendation

In addition, we have enriched input types, increasing voice recognition. We invocate the Baidu voice recognition. The voice recognition services expand the types of data, and we can process and improve the interactivity and data processing capabilities of chatbots. **Figure 9** and **Figure 10** show the code of Baidu voice recognition.

```
""" 你的 APPID AK SK """
APP_ID = '11031096'
API_KEY = 'Znj7ZUGi7HK93nDEGAXdzAjI'
SECRET_KEY = '64c3e4a4ba912a4e0dde68fafbea127e'

# 19430124 FAN Shuaishuai 's redis
HOST = "redis-13670.c8.us-east-1-4.ec2.cloud.redislabs.com"
PWD = "uOuqljWUzWvmfoDuf6CRsonlrfUmYKSD"
PORT = "13670"

client = AipSpeech(APP_ID, API_KEY, SECRET_KEY)
redis1 = redis.Redis(host=HOST, password=PWD, port=PORT, decode_responses=True)


def get_module_dir(name):
    path = getattr(sys.modules[name], '__file__', None)
    if not path:
        raise AttributeError('module %s has not attribute __file__' % name)
    return os.path.dirname(os.path.abspath(path))
```

Figure 9. shows the code of Baidu voice recognition

```python
# Handler function for File Message
def handle_AudioMessage(event):
    message_content = line_bot_api.get_message_content(event.message.id).content
    res = client.asr(
        message_content, 'm4a',
        16000, {
            'dev_pid': 1737,
        })
    line_bot_api.reply_message(
        event.reply_token,
        TextSendMessage(text=alice.respond(res['result'][0]))
    )
```

Figure 10. shows the code of Baidu voice recognition

Conclusion:

Up to now, we complete the following function: daily conversation, the location information and map of COVID-19 patients, COVID-19 news recommendation. We also enrich the input types. The chatbot has two types of queries: content and voice. And the chatbot gives different types of responses: content, Maps and news links, and pictures. We also plan to implement medical knowledge recommendations, health knowledge question and answer.

The **social impact in real world** of this project is the functions mentioned above. The chatbot is a companion, information recommender, and personal doctor. Through this chatbot, the users can get some information and news related to the COVID-19, which makes some contributions to anti-epidemic viruses. In addition, we have made some contributions to the public health vertical.

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

From small businesses to global enterprises, the cloud is a very hot topic, it is a very broad concept, covering many online areas. Cloud services are just a general term, which can be divided into three categories (IaaS, PaaS, SaaS). Infrastructure as a service (IaaS), Platform as a service (PaaS), Software as a service (SaaS). IaaS provides cloud consumers with a high level of control and responsibility for the configuration and utilization of IT resources. PaaS is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app. SaaS is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. **Figure 11** shows the difference between IaaS, PaaS, and SaaS.

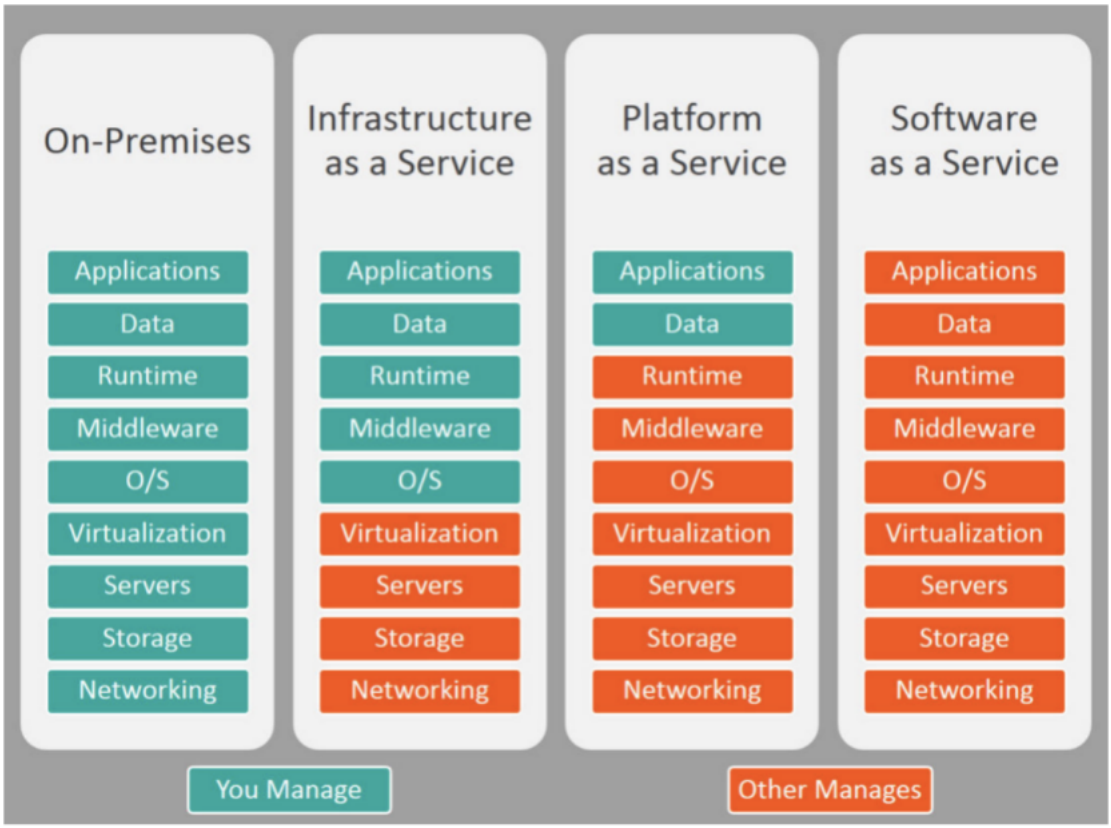

Figure 11. shows the difference of IaaS, PaaS and SaaS.

In this project, we apply Heroku into cloud platform function. Heroku is a cloud platform as a service **(PaaS)** supporting several programming languages. **Figure 12** and **Figure 13** describe the flow chart of Heroku. Seeing a flowchart from HeroKu's official website, it demonstrates the application building channel it advocates, including individual developers, entrepreneurial teams, and even businesses of all sizes can use it in its own way, and the rest is left to users go to develop excellent applications. Between the developer and the user of the application, there are links such as application deployment, management, and scaling. Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages. In addition, in order to obtain a map of the distribution of COVID-19 patients, we also apply the API service from Autonavi Map. The service provided by Autonavi Map system is **SaaS**. We generate maps by invocating applications on remote servers. SaaS applications are also known as Web-based software, on-demand software and hosted software. End users request services from the cloud as needed, without having to maintain any infrastructure or software operating environment locally.

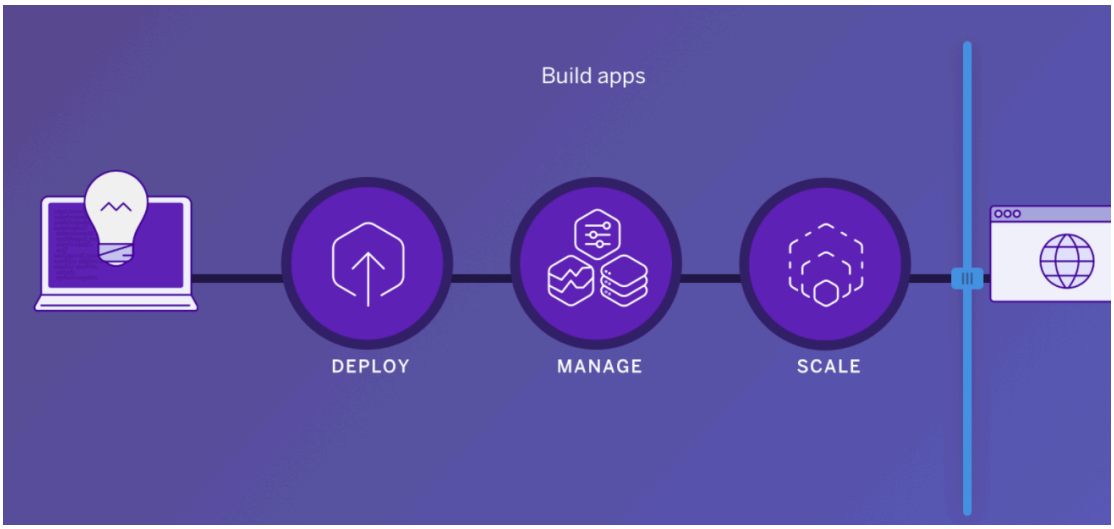

Figure 12. the flow chart of Heroku

Figure 13. the flow chart of Heroku