

## Milestone 4 - Project Presentation and Documentation

Due date: 15th Apr, 2020 (Wednesday)

Marking Criteria: 10% for milestone4.pdf. 0% if the document is severely late.

Answer the following questions:

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

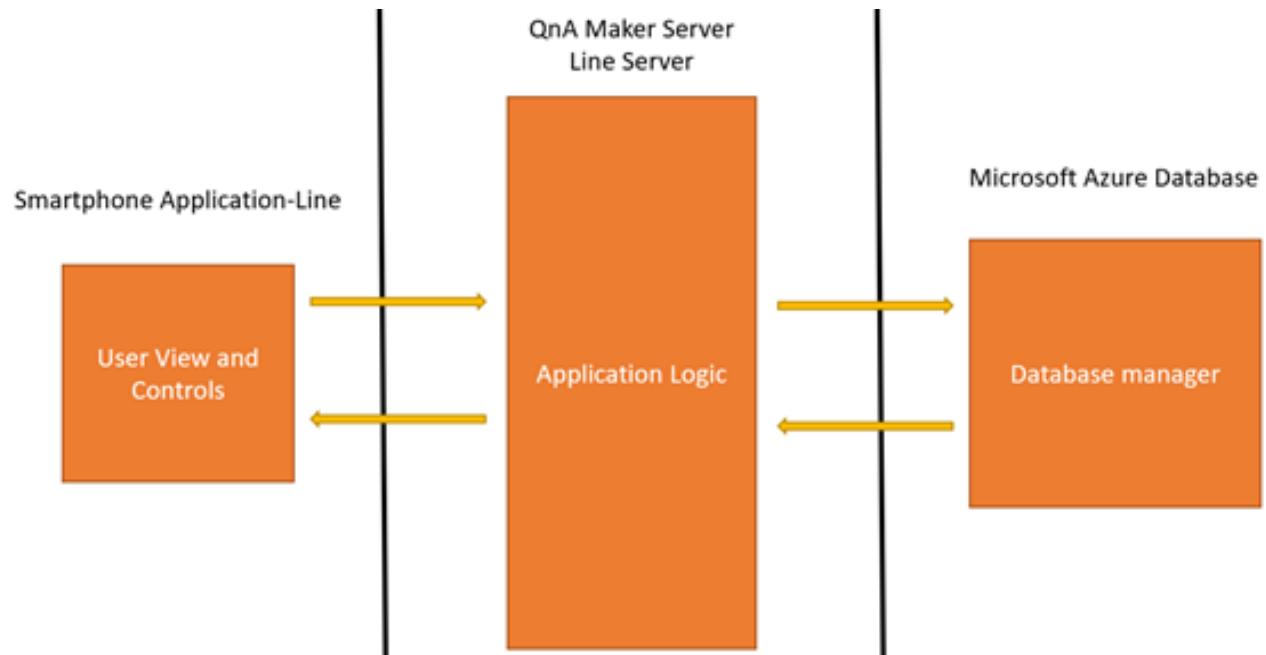
Architectural elements:

We set all the search keywords and answers on QnA Maker. QnA Maker connected with the Line Chatbot through Webhook. Since Webhook worked through request reply protocols, the communication paradigm of our chatbot is remote invocation and the architectural styles is Client-server. In addition, the chatbot was launch in Line application, which was convenience for users search the latest Covid-19 information on smartphone, thus we the utilizes the mobile code for placement.

Layering:

Application/Service	Chatbot – enquire about the info of Covid-19
Middleware	QnA Maker and Line Message API
Operating System	IOS/Android
Computer and network hardware	Smartphone and Network

## Three tier Architecture



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

To increase the capacity of the chat bot service, we setup several tools for accessing and monitor the usage of our bot, then determine what aspects that need to improve on. After accessing both Line developer panel and Microsoft Azure platform, the former one only provides daily traffic in statistical review, while the later provides a variety of software and tools for monitoring. For instance, we are using Azure platform and setup several sets of queries to monitor the bot's traffic.

[illegible]

The first type of queries we used show a general log. It records each question entry with details such as time, wordings used in the questions, answer replied, latency, coefficient score between the users wording and the wording we set on the database etc. These data help us to improve the capacity of the bot such as determine what questions we may have missed in the database yet popular in the cloud consumer side.

Run

Time range : Set in query

Save

Copy link

New alert rule

Export

```
//Total Question Traffic in a given time period
let startDate = todatetime('2020-03-01');
let endDate = todatetime('2020-04-13');
requests
| where timestamp <= endDate and timestamp >= startDate
| where url endswith "generateAnswer" and name startswith "POST"
| parse kind = regex url with "(?i)knowledgebases/"KbId"/generateAnswer"
| summarize ChatCount=count() by KbId
```

Run

Time range : Last 24 hours

Save

Copy link

New alert rule

Export

```
//Total Traffic
requests
| where url endswith "generateAnswer" and name startswith "POST"
| parse kind = regex url with "(?i)knowledgebases/"KbId"/generateAnswer"
| summarize ChatCount=count() by bin(timestamp, id), KbId
```

Results

Chart

Columns

Display time (UTC+00:00)

Completed

Drag a column header and drop it here to group by that column

KbId

ChatCount

a814b47c-ba04-4875-93f4-855645333702

124

KbId

a814b47c-ba04-4875-93f4-855645333702

ChatCount

124

Results

Chart

Columns

Display time (UTC+00:00)

Completed. Showing results from the last 24 hours.

Drag a column header and drop it here to group by that column

timestamp [UTC]

KbId

ChatCount

4/13/2020, 12:00:00.000 AM

a814b47c-ba04-4875-93f4-855645333702

46

timestamp [UTC]

2020-04-13T00:00:00Z

KbId

a814b47c-ba04-4875-93f4-855645333702

ChatCount

46

4/12/2020, 12:00:00.000 AM

a814b47c-ba04-4875-93f4-855645333702

4

timestamp [UTC]

2020-04-12T00:00:00Z

KbId

a814b47c-ba04-4875-93f4-855645333702

ChatCount

4

Run

Time range : Last 24 hours

Save

Copy link

New alert rule

Export

```
//User Traffic
requests
| where url endswith "generateAnswer"
| project timestamp, id, url, resultCode, duration
| parse kind = regex url with "(?i)knowledgebases/"KbId"/generateAnswer"
| join kind=inner {
| traces | extend id = operation_ParentId
} on id
| extend UserId = tostring(customDimensions['UserId'])
| summarize ChatCount=count() by bin(timestamp, id), UserId, KbId
```

Results

Chart

Columns

Display time (UTC+00:00)

Completed. Showing results from the last 24 hours.

Drag a column header and drop it here to group by that column

timestamp [UTC]

UserId

KbId

ChatCount

4/13/2020, 12:00:00.000 AM

a814b47c-ba04-4875-93f4-855645333702

23

timestamp [UTC]

2020-04-13T00:00:00Z

KbId

a814b47c-ba04-4875-93f4-855645333702

ChatCount

23

4/12/2020, 12:00:00.000 AM

a814b47c-ba04-4875-93f4-855645333702

2

timestamp [UTC]

2020-04-12T00:00:00Z

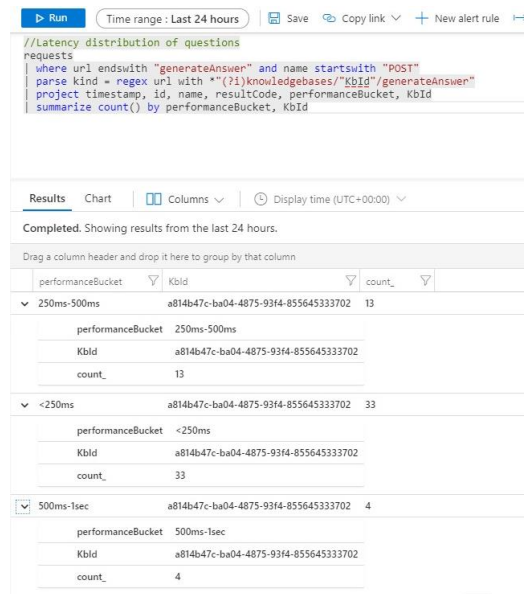
KbId

a814b47c-ba04-4875-93f4-855645333702

ChatCount

2

The second types of queries show the chat-count over a period of time. We can view the traffic on a daily interval or any given period. By knowing how many questions the bot had computed and answered, we can project the future traffic according the tendency of the past record. By changing the criteria of the query, we can single out whether the questions that asked “correctly”, which get responded with answer in the database, or not. We could know if the database is effective or need more accurate question keywords and answers.



The third type of queries show the general latency of questions, we can monitor the bot if too many questions are responded in high latency. It could mean the bot takes more time to identify the keywords in the questions. We can optimize the database by better defining the question words to shorten the respond time. It helps to improve the general capacity of the bot.

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

Our bot can be identified as a SaaS (Software as a Service), it is a bot service using Line API to interact with the cloud consumer and use Microsoft Azure QnA maker to provide necessary information to the Line API.

In Cloud consumer's perspective, they are limited to front-end usage only, which provides QnA chat and receive autoreply from the bot. Cloud consumers need to install Line Application first, add our bot as friend in order to use the bot. They have no control beyond the designed functions. They cannot develop or setup additional functions to the bot.

In developer perspective, the overall project developments are based on pre-established platforms (Line, Microsoft Azure). We as the developer do not practically own any physical hardware nor IT resources to create and maintain this bot services. Moreover, the usage of the cloud consumers is monitored via the Line and Azure platforms.