Thanh Danh Phan

Customer service enhancement

Data Mining and Cognitive System Approach

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Degree Program: Information Technology

8 November 2017

Author(s) Title Number of Pages Date	Thanh Danh Phan Customer service enhancement - Data Mining and Cognitive System Approach 52 pages + 2 appendices 8 November 2017
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialization option	Smart System
Instructor(s)	Olli Hämäläinen, Senior Lecturer, Metropolia UAS Zhongliang Hu, Project Manager, ABB

This decade has seen an enormous growth in the amount of data. The more data is analyzed, the better customer insights are gained. The ultimate goal is to make customers satisfied, which means interpreted data should be used efficiently to make products better accordingly.

In this thesis, different solutions and services were compared by showing their advantages and disadvantages. Performance and scalability were covered briefly since there is not only a need to make the web application stable to simultaneously serve hundred customers but also an opportunity to expand the project.

Desk research technique and qualitative approach were employed to collect user feedback and requirements. Different processes, such as Cross Industry Standard Process for Data Mining and Continuous Integration Process, were followed. In addition, Angular 4 and .NET Core were chosen to build the application.

The outcome of the study includes a functional web application for data analysis and data prediction as well as a new cloud service, which improves the current customer service, by using text analytics.

Keywords	Analytics, cognitive system, data mining, natural language processing, time series, web application

Contents

Glossary

AI	**A**rtificial **I**ntelligence The field of AI attempts not only to understand but also to build intelligent entities. There are four approaches to AI: thinking humanly, thinking rationally, acting humanly, and acting rationally. **(1)**
ADF	**A**ugmented **D**ickey–**F**uller test An augmented version of original Dickey-Fuller test and is capable of performing a test for larger and more complicated set of time series models.
API	**A**pplication **P**rogramming **I**nterface API includes different specifications and exists in many forms. It allows different programs to communicate with each other.
ADFS	**A**ctive **D**irectory **F**ederation **S**ervices ADFS provides access control across a wide variety of applications including Office 365, cloud based SaaS applications, and applications on the corporate network. (2)
ARIMA	**A**uto-**r**egressive **I**ntegrated **M**oving **A**verage ARIMA model is a data model, which is used widely in time series analysis to give a better data understanding and predict future data points.
Angular	A Typescript-based web application platform, developed by Google.
Angular CLI	Angular **C**ommand **L**ine **I**nterface Angular CLI contains required packages and settings for a project, which makes development process become easy. Available at: https://github.com/angular/angular-cli
Angular Material	An UI Component framework based on Google's Material Design. Available at: https://material.angular.io/
Bootstrap CSS	A popular library for developing responsive and mobile-first web sites. Available at: https://getbootstrap.com/css/
Business logic	A part of a program, which encodes the real-world business rules that determine how data can be created, stored, and changed.

CI	**C**ontinuous **I**ntegration CI is the process of automating the build and testing of code every time changes are committed to version control (3)
CORS	**C**ross-**O**rigin **R**esource **S**haring A mechanism to secure web resources.
Chrome	A web browser which is developed by Google. Available at: https://google.com/chrome/browser/desktop
CRISP-DM	**C**ross **I**ndustry **S**tandard **P**rocess for **D**ata **M**ining CRISP-DM is a result of the cooperation from over 200 organizations interesting in using data mining internally or promoting use cases of data mining.
DRY	**D**on't **R**epeat **Y**ourself A software development principle, which aims at reducing repetition of software patterns.
Font Awesome	Font Awesome is icon library based on CSS. Available at: https://fontawesome.io/
Google Material Design	A design language, which was first announced in 2014 by Google, can be found on most of Google's product. Available at: https://material.io/guidelines/
HTTP	**H**yper**t**ext **T**ransfer **P**rotocol The foundation of Word Wide Web. It was invented by Tim Berners-Lee and his team at CERN.
IBMWNLU	**IBM W**atson **N**atural **L**anguage **U**nderstanding An IBM natural language processing service, which is used for advanced text analytics.
JSON	**J**ava**S**cript **O**bject **N**otation A syntax for storing and exchanging data.
MATA	**M**icrosoft **A**zure **T**ext **A**nalytics A Microsoft natural language processing service, which is used for advanced text analytics.

NLP	**N**atural **L**anguage **P**rocessing Natural language processing enhances the ability of machines to understand human language. It can be applied in speech recognition, natural language understanding, natural language detection and many more.
NPM	A package manager for JavaScript Available at: https://www.npmjs.com/
Protractor	A test framework for Angular applications. Available at: http://www.protractortest.org
RMSE	**R**oot **M**ean **S**quare **E**rror RMSE is used to measure of the differences between observed values and the ones generated by data model.
SignalR	A .NET library, which supports real-time web functionality. Available at: https://www.asp.net/signalr
Sublime Text	A widely used text editor, which is written in C++ and Python, Available at: https://www.sublimetext.com/
Typescript	An open source programming language developed by Microsoft.
UI	**U**ser **i**nterface UI is the element where user and software interactions occur.
Visual Studio	An integrated development environment from Microsoft. Available at: https://www.visualstudio.com/downloads/
Webpack	An open source package which is used for module bundler. Available at: https://webpack.github.io/

1 Introduction

> "The goal is to transform data into information and information into insight."
>
> – Carly Fiorina - Information: the currency of the digital age (4)

This decade has seen an enormous growth in the amount of data. In IBM 2013 Annual Report, the company stated that the world was generating more than 2.5 billion gigabytes of data every day (5); Facebook generated 4 new petabytes of data and ran 600,000 queries per day in 2014 (6); just YouTube Spaces alone produced over 10,000 videos, which generated over 1 billion views as of March 2015 (7). New data is inadvertently created not only by what people do on the Internet (such as adding a new comment on LinkedIn, clicking a "like" button on Facebook) but also by their physical behavior such as subscribing to a magazine or buying a train ticket. These data may contain a pattern of user characteristics, their interests and life style. Data mining allows experts to interpret those patterns and extract essential information from given data set. By doing so, service providers can make correct decisions and improve their services based on customer needs.

Secondly, natural language processing (NLP), a sub-set of Artificial Intelligence, is one of the most exciting technologies where computers can extract information from given input in many forms, such as written documents and voice records. With noticeable evolution, NLP has become a promising candidate to improve product service by studying user responses. More and more companies have been using NLP to improve the quality of their products and to provide user an optimal support. For example, Apple's Siri, an intelligent personal assistant first appearing in October 2011, has been integrated to macOS and iOS to enhance the capability of voice command.

Thirdly, the case study company in this thesis is developing a real-time chat application based on the needs of customers. This means service team now will be able to enhance user experience and serve users from anywhere in the world just in a nick of time. However, there are two problems which need consideration. The first problem relates to establishing a connection between the support team and customer. By the time of this writing, the chat application does not have an ability to connect a customer to a correct support user with suitable domain knowledge. The second problem is a lack of a tool to analyze the application mentioned above. Without an analyzing tool, the support team

will be unable to measure the performance and the effectiveness of the product. These are only two possible problems that may occur and if they are not solved properly, together they can reduce customer satisfaction and increase customer attrition.

Consequently, the combination of data mining and NLP is likely to be the best solution for the above-mentioned problems. By using IBM Watson Natural Language Understanding (IBMWNLU) to analyze problem description sent by customer, a support user with related knowledge will be assigned to support that specific customer. Data will be analyzed and data trends can be predicted by modelling data to give the team a general view of customer needs.

With the business challenge in mind, the study aims to answer the following questions:

> How can data mining and natural language processing improve customer service? Is it easy to integrate them to developing application?

The outcome of the study includes a functional web application for data analyzation and data prediction as well as a new cloud service, which improves the current customer service, by using text analytics. In particular, it should help to free the resource of developers to spend their time with new product development and reduce the cost of maintaining support teams.

In this thesis, different solutions and services were compared by showing their advantages and disadvantages. Performance and scalability were covered briefly since there is not only a need to make the web application stable to simultaneously serve hundred customers but also an opportunity to expand the project.

This study is written in 9 sections. The first three sections will give the reader a general view and necessary information for the project. Section 4 focuses on application architecture while section 5 describes the implementation. Section 6 and section 7 contain information relating to testing and operating cost, respectively. Finally, section 8 and 9 will be for the result and the conclusion.

2 Methods and tools

This section includes information relating to methodologies followed during this study and workflow processes. In addition, it will give information about tools, which were used for the development.

2.1 Project approach

As the objective of this study is to enhance customer service by applying both data mining and NLP, the best way to examine the result is to collect user feedback. In addition, desk research technique was selected because not only the problems were clear but also all necessary documentation could be easily accessed from company intranet. Furthermore, qualitative approach was employed since the mentioned application was still under development and only the most important customers and managers were invited to use the beta version. Therefore, a survey, which targeted said personnel, was created to gain their insights and requirements.

2.2 Workflow process

Throughout the study, three processes were followed: obtaining project requirement, exploring data process and continuous integration process. The first process, getting project requirement, is based on “The process of qualitative analysis” developed by Christine, Immy and Matt (8). Because no programming is required in this phrase, the process is modified to adapt the context and is illustrated as figure 1.

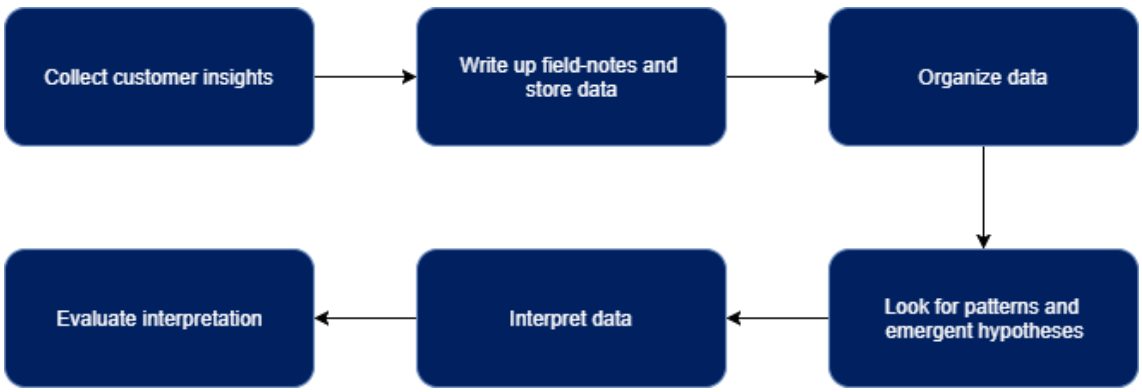

Figure 1. Getting project requirement process

Secondly, Cross Industry Standard Process for Data Mining (CRISP-DM) (9) is primarily used to handle the data generated by mentioned application. This process includes 6 different phrases as shown in figure 2.

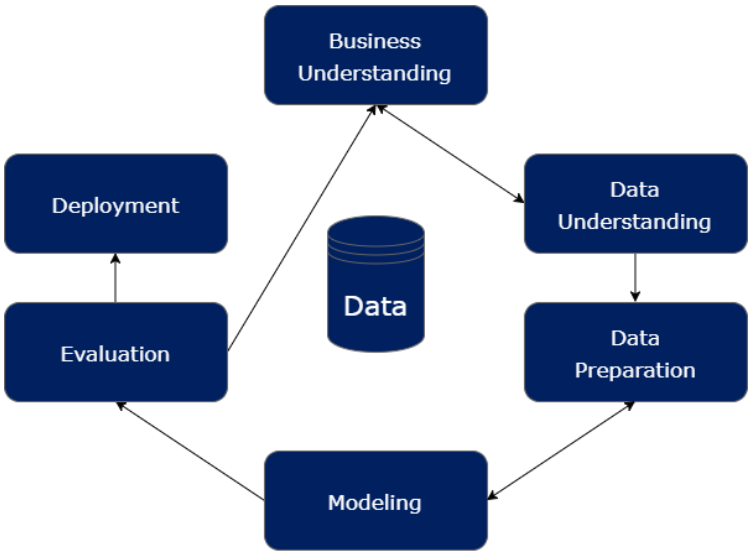

Figure 2. CRISP-DM process

The CRISP-DM is a result of the cooperation from over 200 organizations interested in using data mining internally or promoting use cases of data mining (10). It was built based on the key idea that “something could be applied independent of any certain tool or kind of data” (11).

Finally, Continuous Integration (CI) is used to manage the whole project. The process is shown in figure 3.

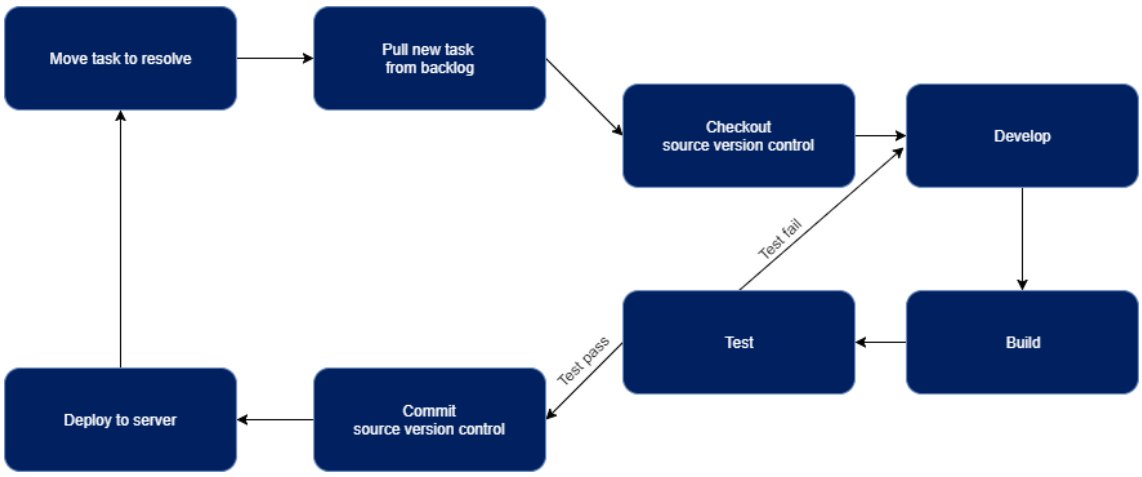

Figure 3. CI process

As shown in CI process diagram, a new code block will only be merged to master branch when it passes a test set. Consequently, existence of a steal code block, or non-working one, will be kept to a minimum in the master branch.

2.3 Technology

There are many front-end frameworks to choose from these days depending on the needs. At the end, Angular 4 was chosen since it can be easily scaled up and has extreme performance. In addition, the front-end also used Font Awesome for the icons and Bootstrap CSS for styling elements. Thus, needed package for front-end development was managed by NPM. To optimize the application, Webpack was used to bundle and minify the code.

The back-end followed WISA Stack (Window – IIS – SQL Server and ASP. NET). Therefore, it was written on C# and used .NET Core framework. Cutting edge technologies with high performance and good scalability is the reason why .NET Core was chosen instead of other .NET frameworks. The whole project will be later deployed to Microsoft Azure.

Finally, Python was the chosen language to build data model, analyze data and make prediction. In addition, Python is an easy to learn language with huge number of libraries for data-exploring purpose.

2.4 Tools

The project was divided into multiple phases and different phase required different tools. During the first phase, which was implementing front-end, macOS and Sublime Text had been used for code editing.

Since the project will be hosted on Microsoft Azure Cloud and the back-end was written in C#, Windows was the main operating system for all other phases. Consequently, Visual Studio 2015 was used for code editing and deploying the app to the cloud. Debugging and performance recording were based on Chrome Inspection.

For illustrating purpose, Jupyter IDE was used while exploring data with Python. In addition, Anaconda distribution, one of the most popular Python data science platforms, may

worth considering since it automatically installs Python, SciPy, NumPy and other necessary data science libraries.

In management point of view, it is good to have places where work can be tracked daily and code can be stored. Because the project was developed only by one person, there was no need to use a complex solution such as Microsoft Team Foundation Server. Therefore, GitHub with private access for version control and Trello for task management were used.

3 Theoretical background

Before using IBMWNLU APIs, analyzing data and making predictions, some theories relating to NLP and time series are needed. Foundationally, IBMWNLU is an application of advanced NLP, Information Retrieval, Knowledge Representation and Reasoning, and Machine Learning technologies (12). On the one hand, as the scope of this project is to use ready-made APIs provided by IBMWNLU to identify language and extract keywords, only NLP is described in this section. On the other hand, data analysis and prediction are made from scratch. Consequently, all necessary theories relating to processing time series data are needed.

3.1 Natural language processing

By processing natural languages, computers are able not only to communicate with humans but also extract information from written language. Human languages are ambiguous with complex grammar and lexical diversity. In addition, a single sentence may have different meanings depending on the context and languages are constantly changing. The mentioned properties make natural languages become difficult for machines to study.

Furthermore, it is estimated that more than 80 percent of the world data is unstructured, which means data does not have a pre-defined data model or is not organized in a pre-defined manner (13). Consequently, for machines to process natural languages, language models are needed and by using them, language identification, spelling correction, and genre classification can be done. (1)

3.1.1 Language models

A written language is presented as a combination of numerous words and each word is constituted by multiple characters. Therefore, a probability distribution over sequences of characters is one of the simplest language models. A word, which has a length of n characters, is called a n-gram, such as 1-gram (unigram), 2-gram (bigram) and 3-gram (trigram). In addition, n-gram accepts a character, word, syllable and *others* as unit; they

are called n-gram character-level, word-level, syllable-level and *others*-level, respectively.

The Markov process (Markov chain), named after the Russian mathematician Andrey Markov, is a stochastic (imperfectly predictable) process which current state depends on only a finite fixed number of previous states (1). Figure 4 and figure 5 illustrate how each state relates to others in first-order and second-order process, respectively.

Figure 4. A first-order Markov process

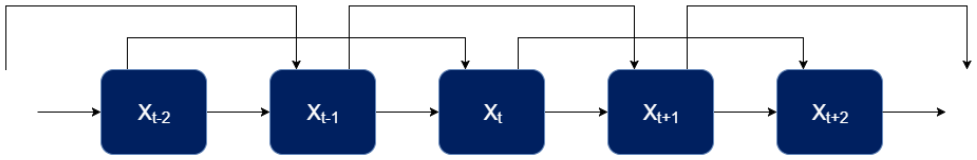

Figure 5. A second-order Markov process

A n-gram model is defined as a Markov chain of order $n-1$. For example, a trigram model, which is a Markov chain of order 2, is written as:

$$P(c_1|c_{1:i-1}) = P(c_i|\, c_{i-2:i-1})$$

Therefore, $P(c_{1:N})$, a notation of a probability of sequence having N characters from c_1 to c_N, under trigram model can be inherited and written as:

$$P(c_{1:N}) = \prod_{i=1}^{N} P(c_1|c_{1:i-1}) = \prod_{i=1}^{N} P(c_i|\, c_{i-2:i-1})$$

In practice, probability distributions are smoothed by assigning small non-zero probabilities to unseen words. Consequently, to keep the sum of all probabilities equal to 1, other seen words probabilities are slightly decreased. Smoothing techniques are various, from a simple one such as Laplace smoothing to a more sophisticated one, such as Linear Interpolation smoothing.

Corpus (plural form is corpora) is a large text. Brown Corpus, the first million-word electronic corpus from 500 samples of English text, was created in 1961 at Brown University.

As shown in listing 1 and figure 6, appearance percentages of “the”, “a” and “an” in Brown Corpus are relatively similar to Google Books Corpus.

```
Word: the  Times: 69971    Frequency: 6.025790739171472 %
Word: a    Times: 23195    Frequency: 1.9975163452727887 %
Word: an   Times: 3740     Frequency: 0.3220828252347588 %
```

Listing 1. Selected word frequencies in Brown Corpus

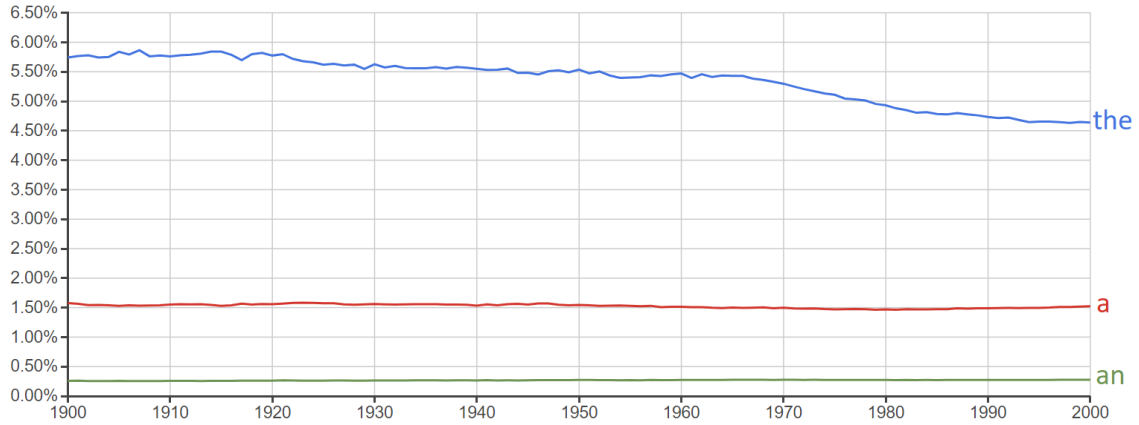

Figure 6. Selected word frequencies in Google Books English Corpus from 1900 - 2000 (14)

Although the number of characters in an alphabet is limited to a specific number, there are unlimited ways to use words formed by those characters. It means the frequency of a particular word may vary through time as illustrated in figure 6.

With linguistic models, computer systems identify languages with greater than 99 percent accuracy; occasionally, closely related languages, such as Swedish and Norwegian, are confused (1). One approach to identify languages is to build a trigram character model of multiple languages with at least 100000 characters for each language.

$$\ell* = \underset{\ell}{\operatorname{argmax}} P(\ell|C_{1:N})$$

$$= \underset{\ell}{\operatorname{argmax}} P(\ell)P(C_{1:N}|\ell)$$

$$= \underset{\ell}{\operatorname{argmax}} P(\ell)\prod_{i=1}^{N} P(C_i|C_{i-2:i-1},\ell)$$

Where $P(c_i|c_{i-2:i-1}, \ell)$ is trigram character model and ℓ is ranges over languages. The most probable language, notated as $\ell *$, is detected by applying Bayes' rule and Markov process assumption.

3.1.2 Information extraction

Information extraction is an automatic process of extracting information from documents. In order to extract information from a given text, a template (pattern) is defined to match the text structure. For example, a general template (not tied to a specific domain) with high precision (almost always correct when they match) and low recall (do not always match) is described as:

$$NP_0 \; such \; as \; \{ NP_1, NP_2, \ldots, (and|or) \} \; NP_n \quad (15)$$

where NP stands for nominal phrase. In addition, to extract information, finding specific relations among extracted words based on text domain is also needed. Considering a sentence: "Tomorrow, the storm front will bring heavy rain to the town", set of relations is time and location; and text domain is weather forecast. As shown in table 1, there are 8 general templates, which cover approximately 95 percent of English text structure.

Type	Template	Example	Frequency
Verb	NP_1 Verb NP_2	X established Y	38%
Noun-Prep	NP_1 NP Prep NP_2	X settlement with Y	23%
Verb-Prep	NP_1 Verb Prep NP_2	X moved to Y	16%
Infinitive	NP_1 to Verb NP_2	X plans to acquire Y	9%
Modifier	NP_1 Verb NP_2 Noun	X is Y winner	5%
Noun-Coordinate	NP_1 (, | and | - | :) NP_2 NP	X-Y deal	2%
Verb-Coordinate	NP_1 (, | and) NP_2 Verb	X, Y merge	1%
Appositive	NP_1 NP (: | ,) NP_2	X hometown: Y	1%

Table 1. 8 general templates that cover about 95 percent of the ways that relations are expressed in English. (1)

Consequently, in narrowly restricted domains, information extraction can be done with high accuracy. The more general the domain gets, the more complex language models

and advanced techniques are needed. Therefore, an extraction system, which is relations-independent, has the ability to read on its own and build up its own database is ideal.

3.2 Time series

A time series is a sequence of data with index as a specific time and is sorted in time order. Normally, time series is a collection of discrete-time data.

> A time series is a set of observations x_t, each one being recorded at a specific time t. A discrete-time time series (the type to which this book is primarily devoted) is one in which the set T_o of times at which observations are made is a discrete set, as is the case, for example, when observations are made at fixed time intervals. Continuous-time time series are obtained when observations are recorded continuously over some time interval, e.g., when $T_o = [0,1]$. (16)

Examples of time series are population of Helsinki city over years, heights of ocean tides and measurements of the annual flow of the Nile river at Aswan. Time series data model may exist in many forms and represent different stochastic processes.

3.2.1 Time series with zero mean

This is the most basic model of time series, which is a sequence of independent and identically distributed (i.i.d) random variables with zero mean. It can be written as

$$\{X_t,\ t = 0, \pm 1, \pm 2, \dots\} \mid E(X_t) = 0.$$

One example of zero mean time series is i.i.d noise.

$$X_t = r_t$$

$$E(X) = \sum_{t=0}^{n} r_t * P(r_t) = 0$$

where r_t is a random variable at time t

Another example of this type of model is binary process. With $x_t \epsilon [0, 1]$, $t \epsilon N$

$$\begin{cases} P(x_t = 1) = p \\ P(x_t = 0) = 1 - p \end{cases}$$

This time series can be reproduced by tossing a coin. Both probabilities of having a head $x_t = 1$ and tail $x_t = 0$ are 50 percent. In both i.i.d noise and binary process, the previous result x_t does not affect and cannot be used to predict the coming result $x_{t+k} \mid k \geq 1$ since the whole data is a sequence of independent and random variables.

3.2.2 Time series with trend and seasonality

In real life, a trend can be easily found from time series data.

$$X_t = m_t + s_t + r_t \mid E(r_t) = 0, m_t = f(t), s_t = g(t), s_{(t-d)} = s_d, d \in N$$

where m_t is a slowly changing function, which acts as trend component; s_t is a period d function, which can be referred to as seasonal component; and r_t is a random variable at time t. As shown in figure 7, it is a clear incidence that the data has an increasing trend over the times.

Figure 7. Monthly number of employed persons in Australia from Jan 1983 – Dec 1990. (17)

In addition, some seasonal variation is also shown in the graph as the number of employed persons tends to follow a similar pattern.

3.3 Stationarity of time series

Stationarity of time series can be loosely described as: a time series $\{X_t,\ t = 0, \pm1, \pm2, \dots\}$ with $E(X_t^2) < \infty$ is said to be stationary if it has statistical properties like time shifted series, which is $\{X_{t+h},\ t = 0, \pm1, \pm2, \dots\} \mid h \in N$. There are two types of stationary time series: weakly stationary time series and strictly (or strongly) stationary series. A weakly stationary time series is when the sequence has constant mean and variance throughout the time and a strictly stationarity time series is when the distribution of a time-series is exactly the same trough time.

> $\{X_t\}$ is weakly stationary if $\mu_x(t)$ is independent of t and $\gamma_x(t+h, t)$ is independent of t for each h. $\{X_t\}$ is strictly stationary if $(X_1, \dots, X_t)' \triangleq (X_{1+h}, \dots, X_{t+h})' \mid h \in N, n \geq 1$. $\triangleq$ is used to indicate that the two random vectors have the same joint distribution function. (16)

where $\mu_x(t) = E(X_t)$ is the mean function of $\{X_t\}$

$\gamma_x(r, s) = Cov(X_r, X_s) = E[\big(X_r - \mu_x(r)\big)(X_s - \mu_x(s))]$ is the covariance function of $\{X_t\} \mid (x, r) \in N$.

3.3.1 Unit root test

A function f is said to have a unit root when $f(1) = 0$. The purpose of unit root test is to determine the stationarity of a time series. Consider a time series

$$X_t = d_t + z_t + \varepsilon_t$$

where d_t , z_t and ε_t is deterministic component, stochastic component and error, respectively. Stochastic component z_t will be tested by unit root test to determine whether it has consequences. There are many unit root tests, such as Phillips–Perron test, Kwiatkowski–Phillips–Schmidt–Shin test, Zivot–Andrews test and Dickey-Fuller test.

3.3.2 Augmented Dickey-Fuller test

Null hypothesis, denoted as H_o, assumes that the stochastic component z_t is non stationary until there is an evidence indicates otherwise. According to the present of a unit root in auto-regressive model, null hypothesis will be accepted or rejected by Dickey-

Fuller test. In addition, it was developed in 1979 and named after the statisticians David Dickey and Wayne Fuller. Augmented Dickey–Fuller (ADF) test, as the name said, is an augmented version of original Dickey-Fuller test and is capable of performing a test for larger and more complicated set of time series models.

$$X_t = c + \theta X_{t-1} + \varepsilon_t \mid E(\varepsilon_t) = 0$$

⇨ $E(X_t) = c + \theta E(X_{t-1}) = \frac{c}{1-\theta}$ (because $E(X_t) = E(X_{t-1})$)

However, this will only valid when $\theta \neq 1$.

$$\Delta X_t = X_t - X_{t-1} = c + \theta X_{t-1} + \varepsilon_t - X_{t-1} = c + (\theta - 1) X_{t-1} + \varepsilon_t$$

$(\theta - 1) X_{t-1}$ acts as stochastic component mentioned in section 3.3.1. Therefore, ADF will check whether $\theta - 1 = 0$ to determine a time series is stationary or not. Listing 2 shows an example of ADF result.

```
ADF Statistic                -3.652342
p-value                       0.004836
Critical Value (1%)          -4.665186
Critical Value (5%)          -3.367187
Critical Value (10%)         -2.802961
```

Listing 2. Example of ADF result of data mentioned in section 5.4

Result from ADF is interpreted by using returned p-value. If the p-value is smaller than 0.05 or even 0.01, the hypothesis that unit root exists (null hypothesis) can be rejected. Another approach is that if ADF statistic, a negative number, is below 0.05 or even 0.01 quantiles, the null hypothesis can also be rejected. In addition, the more negative ADF statistic is, the stronger rejection of the null hypothesis.

3.4 Methods to make time series stationary

The most common causes of stationarity violation are trend and seasonality. For example, as shown in figure 7, number of employed persons keeps increasing because of population growth in the same period, which is illustrated in figure 8.

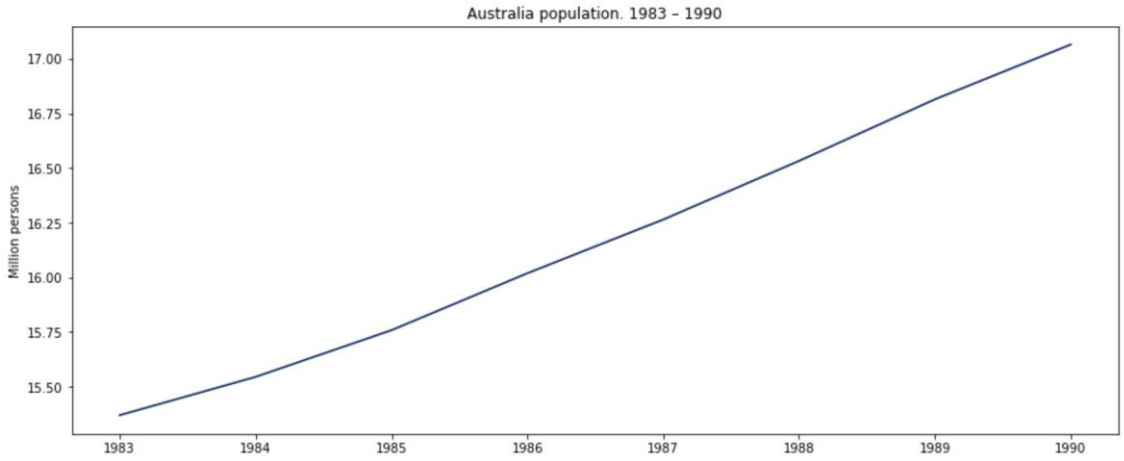

Figure 8. Australia population. Jan 1983 – Dec 1990. (18)

Consequently, by detecting the trend and the seasonality of a sequence and remove them from data may make the process stationary. After modelling the data successfully, the trend and the seasonality will be added back so that the predicted data will have the same property with the original one. There are several methods from basic to advanced to achieve it. For example, the trend can be eliminated by using polynomial fitting, smoothing or data transformation. In addition, decomposition or seasonal difference operator can be used to eliminate both trend and seasonality. However, only the methods used in this project are listed, which are data transformation and seasonal difference operator.

3.4.1 Data transformation

In statistics and mathematics, data transformation means applying a specific function to every single data of the dataset to create a new sequence, which has the same index, with new values.

$$Z_t = f(X_t) \mid \{X_t, t = 0, \pm 1, \pm 2, \dots\}$$

This method is used to stabilize variance. In this sense, the transformation will penalize the high values more than small values. If only positive values are observed, logarithm and square root transformations are usually applied. However, if the set contains both positive and negative values, it is common that a constant will be added to all values to make a new non-negative data set to apply above mentioned transformations. In addition, multiplicative inverse (reciprocal) can be used also in mix of positive and negative case as long as it is a non-zero set. Figure 9, figure 10 and figure 11 illustrate the monthly

number of employed persons in Australia from January 1983 to December 1990 after being applied reciprocal transformation, log transformation and square root transformation, respectively.

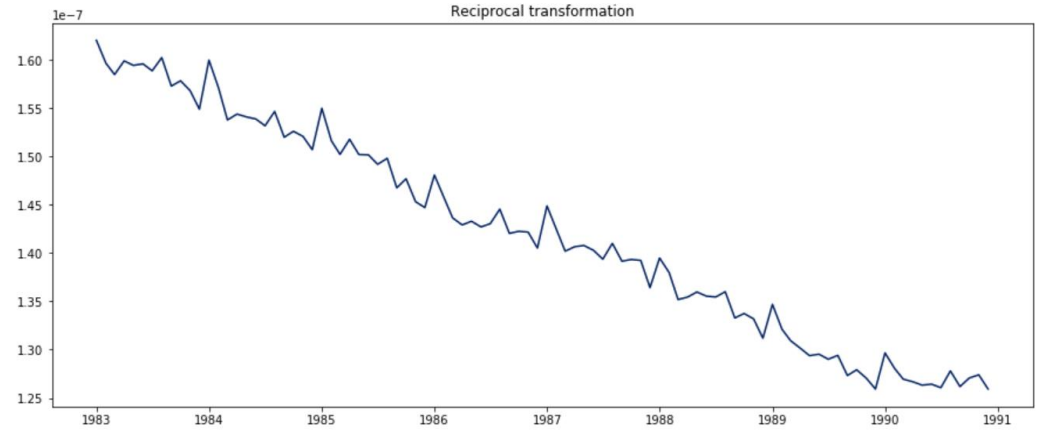

Figure 9. Reciprocal transformation

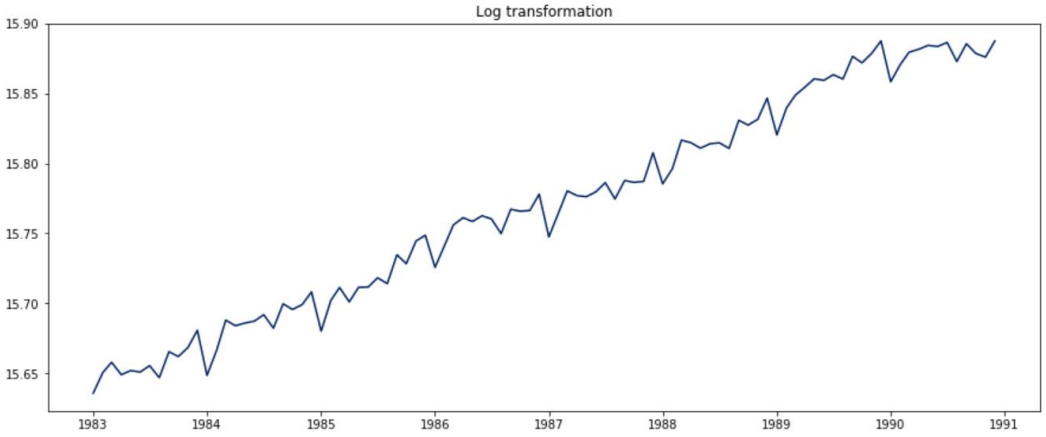

Figure 10. Log transformation

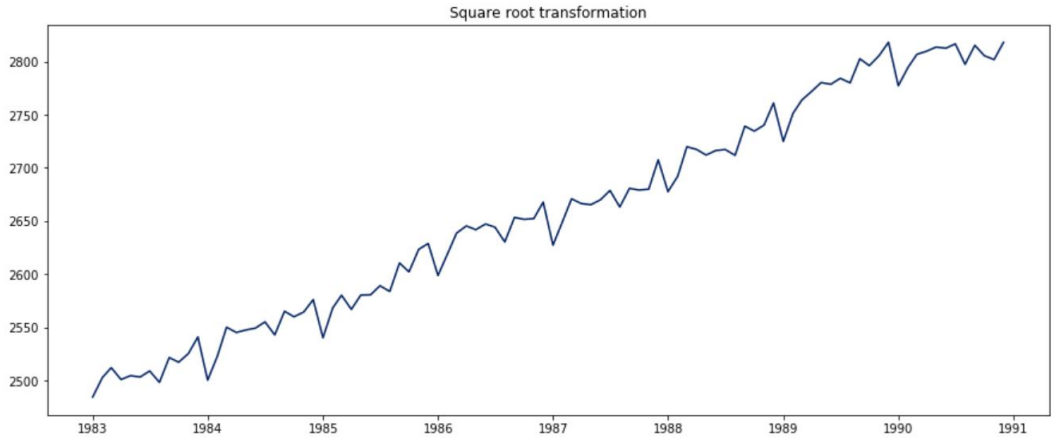

Figure 11. Square root transformation

All though figure 10 and figure 11 have the same shape, log transformation has a faster decay than square root transformation. Take number 1000 as an example, its logarithm is 3 while its square root is 31.6.

3.4.2 Seasonal difference operator

Seasonal difference operator is one of the most commonly used method to eliminate both trend and seasonality. If the cycle of seasonality is known, a new sequence is created by subtracting a specific data to another data which has the same time in the cycle. The formula of first order seasonal differencing is

$$\Delta X_t = X_t - X_{t-n}$$

where n is the cycle of seasonality.

$$\Delta X_t = X_t - X_{t-1} = (1 - L)\, X_t$$

In addition, when n = 1, it will be called difference operator, which is a special case of lag polynomial. Figure 12 shows monthly number of employed persons in Australia from January 1983 to December 1990 after applying first order seasonal differencing where the cycle is 1 month.

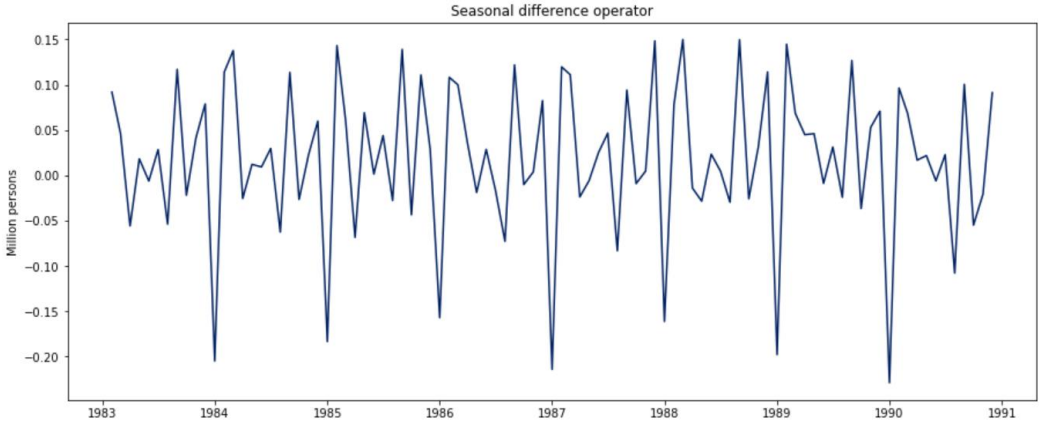

Figure 12. Seasonal difference operator

However, by using seasonal differencing, the data of the first cycle will be lost since there is no prior data with which to do the differencing.

3.5 Auto-regressive integrated moving averages

Auto-regressive integrated moving average (ARIMA) model is widely used in time series analytics. By fitting ARIMA model to given data set, it will not only give a better knowledge relating to mentioned sequence but also enabled the ability to predict future data. A standard notation ARIMA (p, d, d) is used, where all p, d and q are non-negative integers to indicate the specific used ARIMA model. P, d and q are also known as lag order, degree of differencing, and order of moving average, respectively.

3.5.1 Auto-regression process

The auto-regression (AR) process is a process which the next output depends linearly on previous data and a stochastic term. The AR(p) is defined as

$$X_t = c + \sum_{i=1}^{p} \theta_i X_{t-i} + \varepsilon_t$$

where c is a constant, $\{\theta_i, i = i, .., p\}$ are parameters of the model and ε is noise, $\varepsilon \sim N(0, \sigma^2)$. By using notation $\varepsilon \sim N(0, \sigma^2)$, it means data from ε are identically, independently distributed with a normal distribution having mean 0 and the same variance.

The relationship between data in AR process is called correlation. If variables change in the same direction, which means they go up or down together, it is a positive correlation. In case they change in contrary, it is called negative correlation. Otherwise it is called zero correlation. AR model is not always stationary since it may contain a unit root.

3.5.2 Integration

As mentioned in section 3.4.2, differencing is the commonly used technique to eliminate trend and seasonality from a nonstationary time series. Integration (I), which is denoted with I(d), is defined as

$$\Delta^d X_t = (1 - L)^d X_t$$

where d is the times of performing differencing.

3.5.3 Moving-average process

The moving-average (MA) process is a process which the next output depends linearly on current data and various past values of a stochastic term. The MA(q) is defined as

$$X_t = \mu + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t$$

where μ is the mean of the series, $\{\theta_i, i = i, .., p\}$ are parameters of the model and ε is noise, $\varepsilon \sim N(0, \sigma^2)$.

By using lag operator notation polynomial, the MA(q) can be written as

$$X_t = \mu + \theta(L)\varepsilon_t$$

where $L^i y_t = y_{t-i}$ and $\theta(L) = (1 + \theta_1(L) + \ldots + \theta_q(L^q)$. In contrast to AR model, MA model is always stationary since $\theta(L)$ is a finite-degree polynomial. (19)

4 Design

4.1 Architecture

Before developing an application, planning and designing its architecture is a must. Based on the front-end and the back-end selections, the relations of the stack are displayed in figure 13.

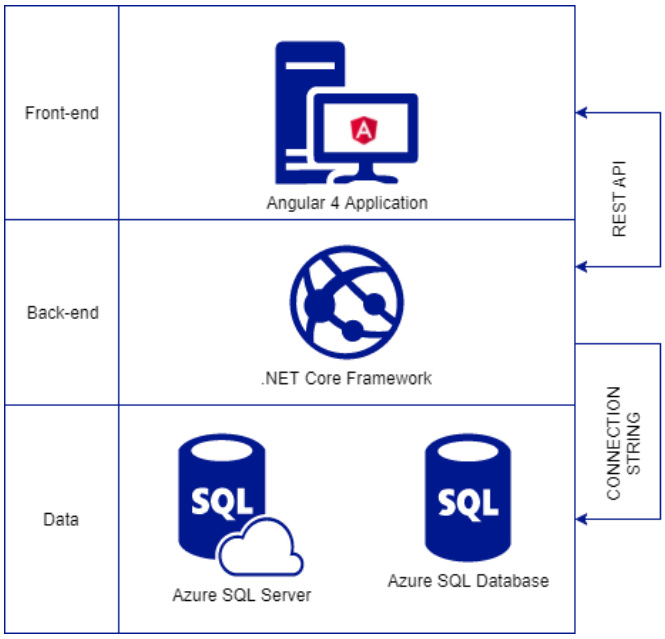

Figure 13. Web application architecture

On the one hand, in .NET Core, Microsoft had unified MVC and Web API Controllers. Therefore, having a back-end, which can host a web application and provide APIs, has become easy. .NET Core connects to Azure SQL Database through “Connection String”. In addition, if an API is called, .NET Core is still able to take over the control and respond to the request instead of redirecting the call to the front-end.

On the other hand, the server should be kept available as much as possible and Angular 4 is a powerful front-end framework. Consequently, most of the logics, view updating and routing are handled by Angular 4.

4.2 Front-end

4.2.1 User interface

The front-end design is based on Google Material Design Guideline. The following question was raised before designing the user interface (UI):

How to make everything simple and clear but still being attractive?

To answer that question, every element should be isolated from each other. Hence, an element is put into a card, which has 3 parts as depicted in figure 14. The top contains the title or the name of the element. The body displays the data (as graph or text). The last part, the utility, allows the user to select the time of the data. In addition, the card is flexible, which means it will automatically expand based on the width of the browser. The primary color is Indigo and secondary color is Teal as shown in the left-hand side of figure 15.

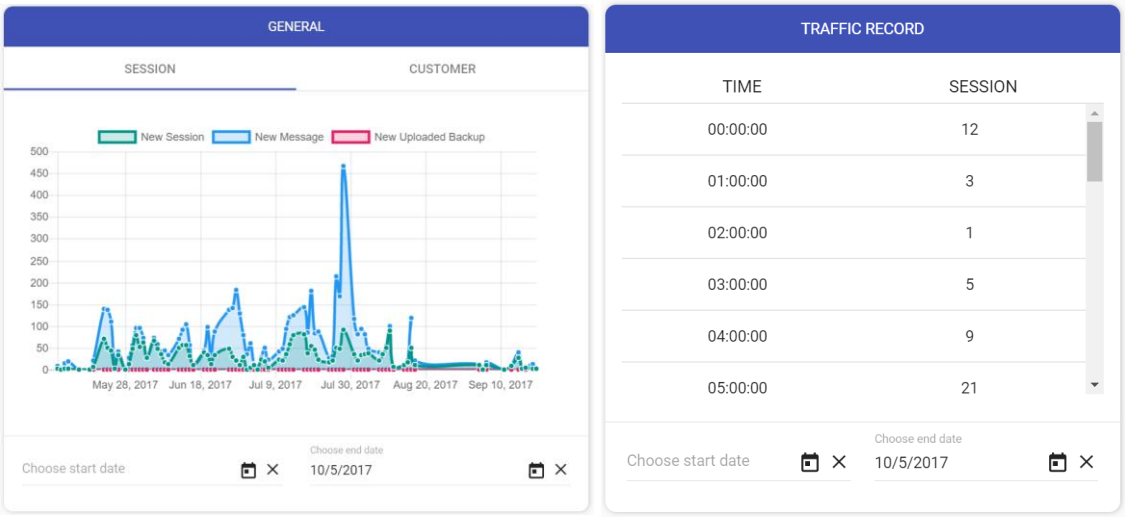

Figure 14. Card design: graph card and text card

Indigo	
500	#3F51B5
100	#B2DFDB
100	#F8BBD0
100	#BBDEFB
100	#FFF9C4

Figure 15. Color schemes

In addition, Pink, Blue and Yellow, which are on the right-hand side of figure 15, are subsidiary colors and are used to display different graph data. Indigo and Teal were chosen to be main colors because they are suitable for long time reading. According to Entrepreneur, blue also stands for stability and reliability (20), which makes users feel comfortable when using the application.

4.2.2 Front-end logic

The front-end needed to be divided into 3 main parts: components, models and services. Every component has its template, which are used to render HTML with CSS and display data models. A model acts as an interface, which helps components access data properties easily. In addition, it also is a bridge, which maps the successful return from API call to valid object which can be used by the front-end. Furthermore, the services are responsible for calling API from the back-end and mapping return data to front-end models. Figure 16 clearly illustrates how the front-end was structured.

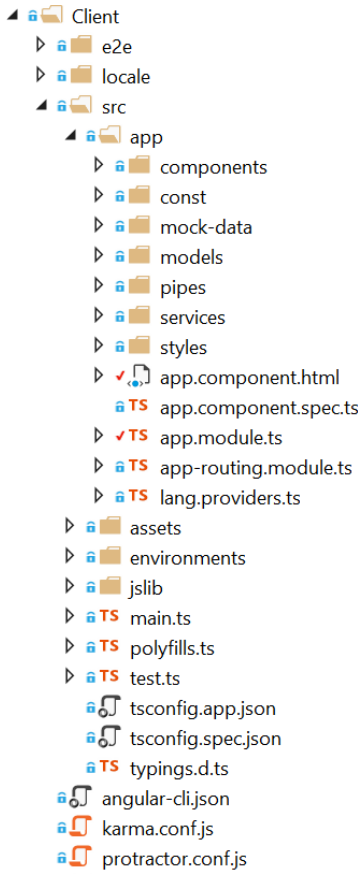

Figure 16. Front-end structure

On the other hand, the routing and authentication were handled in a separate module called app-routing shown partly in listing 3. This module is responsible for checking user credentials and navigate user to correct pages.

```
const APP_ROUTES: Routes = [{
      path: 'login',
      component: LoginComponent
},{
      path: 'dashboard',
      component: DashboardComponent
      canActivate: [AuthGuard]
},{
      path: '',
      redirectTo: 'dashboard',
      pathMatch: 'full'
}, {
      path: '**',
      redirectTo: 'dashboard',
      pathMatch: 'full'
}]
```

Listing 3. Part of app-routing module

Before allowing user to access any components, AuthGuard checks user credential. It would redirect users to login page if they did not login previously. The login checking was put at the master component, which ensures that whatever page is called, the checking is always performed.

4.3 Back-end

4.3.1 Database design

The original database of the project was changed by adding new tables. The relation of them are shown in figure 17. Syslanguage from SQL Server (21), which includes information of 33 languages, can be used for language sorting purposes, which means there is no need to re-create another table to store supported language. Although this table includes information such as dateformat, msglangid, upgrade and many more, only langid and alias (language name in English) are needed. In addition, the reason why alias field is chosen instead of name field is that the name field contains localized language name. For example, Français, which means French in French, is the value of column name and French is the value of column alias.

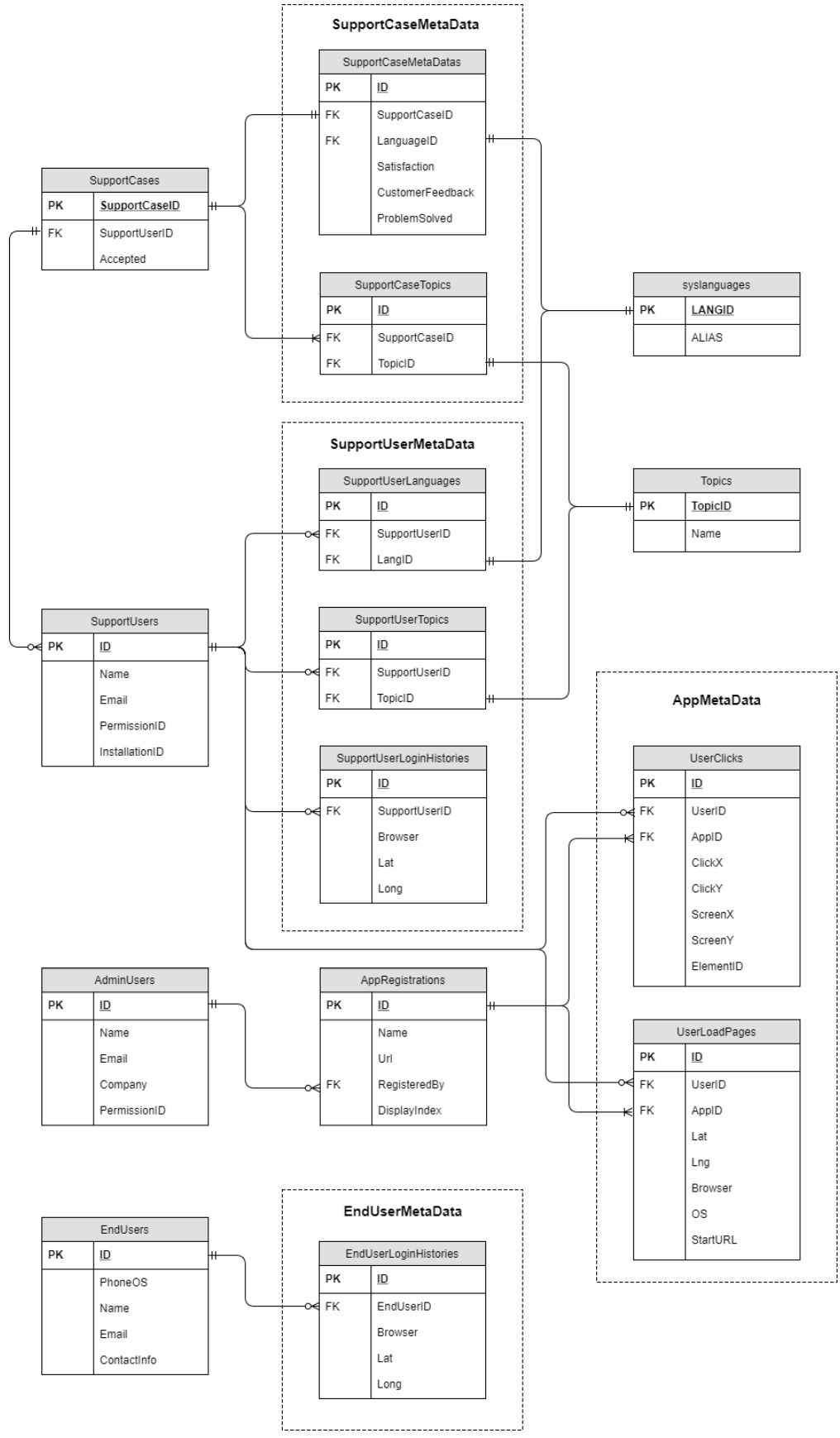

Figure 17. Database data models

By using Support Case Meta Data, the product owner will know the efficient of the application by taking customer satisfaction and customer feedback into account. On the other hand, Support User Meta Data contains necessary information, which helps the mobile application to establish a customer-support user connection based on customer needs and support user expertise. The designer team will be able to improve user experience of the mentioned application by studying support user click behavior and commonly used screen size, which can be extracted from Support User Meta Data.

4.3.2 API design

The back-end handles API calls and returns deserved data. There are two reasons why no authentication was implemented. Firstly, this project was under development phase and did not contain any critical information. Secondly, when this application is integrated to the enterprise production environment, security settings will be configured accordingly based on Active Directory Federation Services (ADFS). As shown in figure 18, every single API call returns a status code accordingly.

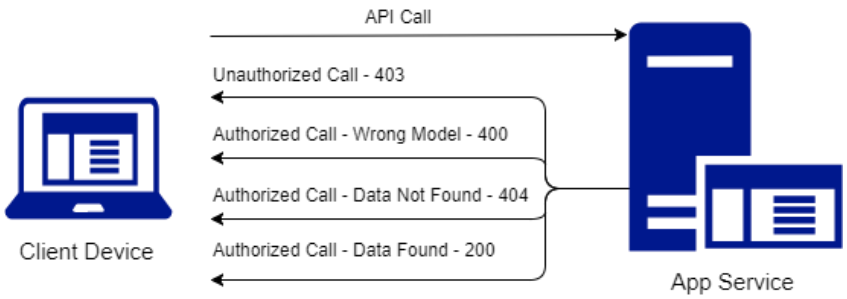

Figure 18. API Handler

In addition, there are more than 5 status codes shown above and they will be handled automatically by .NET Core Framework (22). For example, if the back-end crashes while performing a task in API call, a status code 500, which means server internal error, will be sent to the client.

4.3.3 Text analytics service design

The comparison between IBMWNLU and Microsoft Azure Text Analytics (MATA) shown in Appendix 2 indicates that IBMWNLU is a better service for text analytics purposes. Because of studying purpose, both IBMWNLU and MATA are implemented. Figure 19

and figure 20 illustrates how text analytics services handle problem descriptions sent by the client.

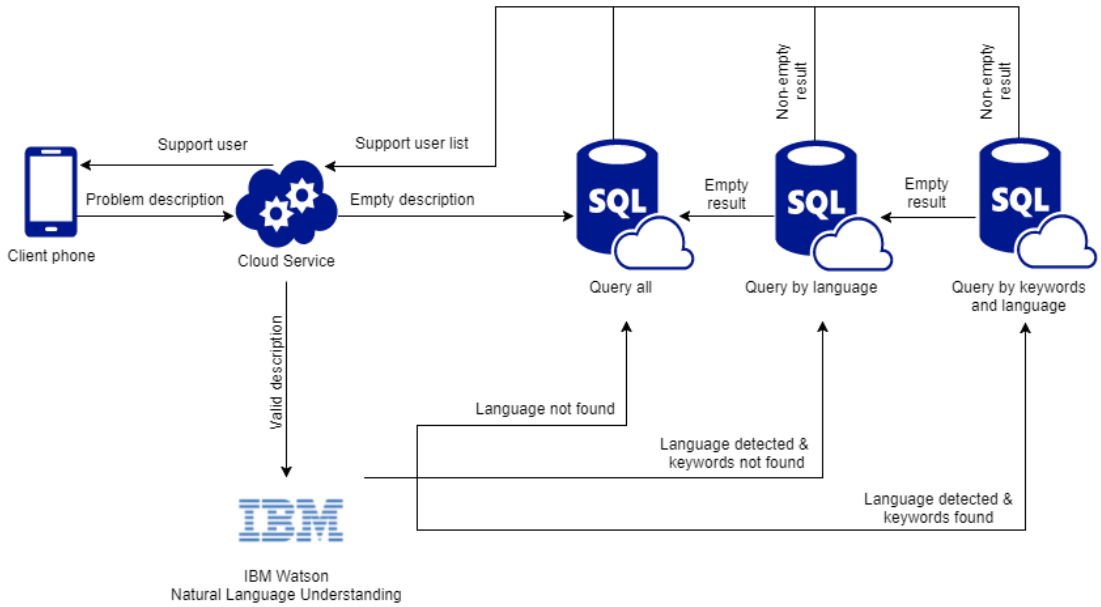

Figure 19. IBM Watson Natural Language Understanding Service

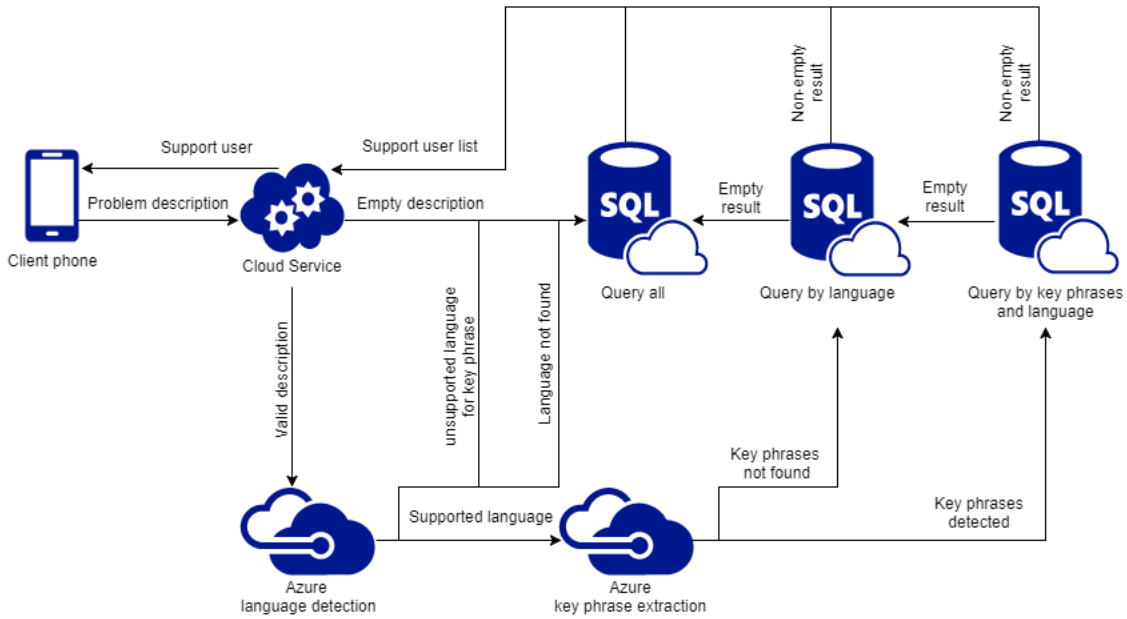

Figure 20. Microsoft Azure Text Analytics Service

The project text analytics service is designed in such a way that it will return a support user in every scenario by using a result, which is returned by cloud services after they

analyze customer problem description, as much as possible. This means it will even query all support users in the worst case when IBM or Microsoft service is not working as expected.

4.3.4 Data prediction design

In order to serve the customers as well as possible and gain more insights of the product, exploring product data is a must. If the data set is qualified, which means noise in the data is insignificant and data set size is big enough, it can be used to build a data model and then predict a new value. In this study, the number of sessions was modelled, which is a time series, and a prediction was made according to that model. They were performed by following the steps in CRISP-DM process, as mentioned in section 2.2, and predicting the data process, as illustrated in figure 21.

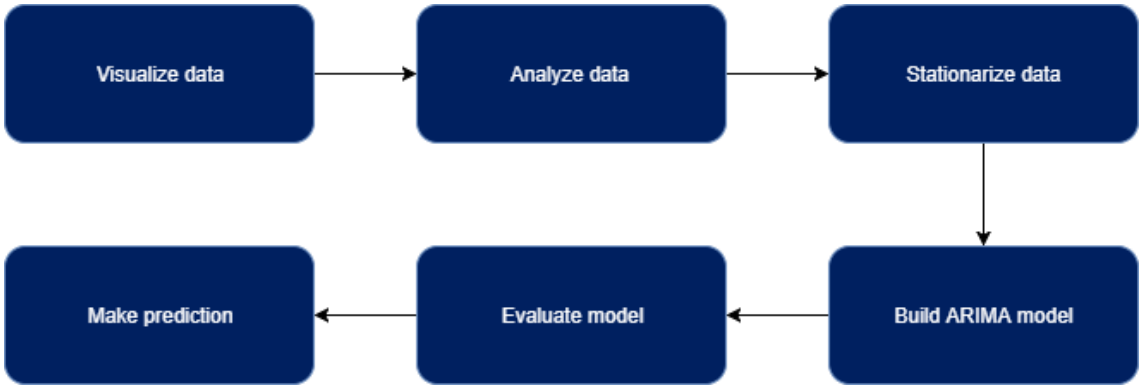

Figure 21. Predicting data process in nutshell

Walk-forward validation was used throughout this section to evaluate candidate models and the final model. A pseudo code of walk-forward validation is shown in listing 4.

```
train_size = time_series_size * 0.5)
train_set = time_series[0:train_size]
test_set = time_series[train_size:]
predictions = pd.Series()
for item in test_set:
    prediction = f(history)
    predictions.append(prediction)
    train_set.append(test_set[i])

rmse = sqrt(mean_squared_error(test_set.values, predictions.values))
print('RMSE: %f' % rmse)
```

Listing 4. Walk-forward validation pseudo code

Initially, 50 percent of the data was used to train model and the other 50 percent was iterated one by one. For each data in the last 50 percent, data model was re-trained based on training dataset, which was the first 50 percent data and previous iteration data, and a prediction was stored for Root Mean Square Error (RMSE) checking.

4.4 Tracking mechanism

In order to analyze websites, tracking mechanism must be considered. It should be convenient for customers to start tracking their websites and guarantee that the track does not affect the performance. In addition, ability of updating without asking customers to explicitly do it manually should be considered.

The solution is a small JavaScript script. This script loads analytics.js, which was also written in JavaScript and hosted on Analytics Web Server. Because the file was hosted on the cloud, more tracking type and functions can be added later without requiring customers to update their websites. This analytics.js tracks user clicks, user browsers, user screen resolution and many more. Then, it posts all collected data, associated with register website number, to Analytics API. Because the analytics.js dynamically reloads every time the user refreshes the page, it must be minimized and optimized for loading speed. However, security must be seriously considered to protect tracked data and to avoid dangerous script injects to customers websites.

5 Implementation

The project implementation was divided into 5 smaller tasks: front-end implementation, back-end implementation, text analytics implementation, data prediction implementation and combined integration.

5.1 Front-end implementation

The front-end was developed on MacOS with the help of Sublime and MacOS Terminal. This was the only task where MacOS was used and it was much easier to set up developing environment on MacOS in comparison to Windows.

5.1.1 Settings

For the sake of simplicity, Angular CLI was used to develop the front-end. Angular CLI seed contains a clear structure and a test template for every single component, which makes the app easy to run unit testing and end-to-end testing. Because Angular application is written in TypeScript, it needs a compiler to transform to JavaScript. This is the place where trouble happens if the compiler configuration file, “tsconfig.json”, is misconfigured. As shown in listing 5, the “target” should be “es5” (ECMAScript 5) and “lib” should be “es2016” (which is “es7”) with “dom” if the application supports Internet Explorer 11.

```
{
      "compileOnSave": true,
      "compilerOptions": {
            "target": "es5",
            "typeRoots": [ node_modules/@types"],
            "lib": [ "es2016", "dom"]
      }
}
```

Listing 5. A part of tsconfig.json

Furthermore, there are three commands can be given to Angular CLI: “start”, “build” and “e2e”. With “start” command and pre-configured dependencies, the project can be hosted on localhost for debugging and refreshed accordingly to code-change in real-time. By giving “build” command to the terminal (later will be replaced with Task Runner Explorer on Visual Studio), the front-end will be chunked and optimized for the best performance.

In addition, the build process is handled by Webpack. A loader needs be specific if there is a need to handle a file type as shown in listing 6.

```
loaders: [
      {
            test: /\.ts$/,
            loaders: ['awesome-typescript-loader', 'angular2-template-loader']
      },
      {
            test: /\.html$/,
            loader: 'html?-minimize'
      }
]
```

Listing 6. A part of Webpack config

The last command, e2e, which means end-to-end, is responsible for triggering test automation. In addition, it simulates end user interaction, which reveals any strange behavior of the application.

As mentioned in section 4.2.1, Angular Material is used to make the UI has a modern looking and a consistent style. It requires a pre-built theme to work, which is Indigo-Pink theme in this project. This theme can be bundled with other CSS files to generate a final CSS file also by Webpack.

5.1.2 Implementation

As mentioned in section 4.2.2, front-end has a clear structure with its components and its services are separate for clarification purposes. While coding, Don't Repeat Yourself (DRY) principle was kept in mind. Therefore, everything was broken down to small modules for later reuse. As shown in listing 7, this "app-container" module receives formatted array data from any component and then displays the given data as graph or text data property.

```
<app-container [title]="deviceTitle"
      [controllers]="deviceController"
      class="col-xs-12 col-sm-12 col-md-4 col-lg-4"
      (timeUpdated)="getEndUserDevice($event)">
</app-container>
```

Listing 7. Example of a component

Event binding, which is timeUpdated in current example, was also implemented. It will emit a signal if the user changes the duration of the dataset and trigger getEndUserDevice function.

5.1.3 Performance

The app needs to handle multiple data transform functions continuously and re-render the UI whenever data is changed. Therefore, performance of the front-end needs to be taken care of seriously.

By default, all elements will be updated and managed by Angular. If an event occurs, for example a data is extracted successfully from an API call, Angular has to check every single component and apply the change accordingly. Consequently, if the data is large and the number of elements are considerable, it will cause an issue to the performance. To make the front-end performs as best as it can, two things need to be done. Firstly, the change detection strategy needs to be configured as shown in listing 8, so elements can only be updated when data is changed.

```
@Component({
    selector: 'app-text-container',
    templateUrl: './text-container.component.html',
    styleUrls: ['./text-container.component.css'],
    changeDetection: ChangeDetectionStrategy.OnPush,
})
```

Listing 8. Change strategy setting

Secondly, given data must be immutable (unchangeable) for change detection. Since the data model is an array, which is mutable (changeable), the services have to return a new array instead of modifying existing one whenever requests are received.

5.2 Back-end implementation

The back-end is written in C# with ASP .NET Core Framework. The reason why it was used is that Microsoft and Microsoft products, such as Microsoft Azure, .NET Framework and many more, easily meet the needs of the enterprise since they have experience of security and integration in enterprise level. From this task onward, Windows will be the main operating system, both locally and on Azure Service.

5.2.1 Implementation

It is a must to use HTTPS by enabling SSL in this application to secure all API call and prevent injection. In addition, all HTTP requests must be redirected to HTTPS and this

can be configured in Startup.cs. As mentioned in section 4.1, .NET Core handled all routes. Consequently, a declaration for the app routes of API and Angular 4 as shown in listing 9 were needed: MapRoute and MapSpaFallbackRoute respectively. However, because the .NET Core just handles URLs relating to API, Angular 4 route module must be aware of unavailable page and redirect the user to the main homepage automatically.

```
app.UseMvc(routes => {
      routes.MapRoute(
            name: "default",
            template: "{controller=Home}/{action=Index}/{id?}");

      routes.MapSpaFallbackRoute(
            name: "spa-fallback",
            defaults: new {controller = "Home", action = "Index"});
      });
}
```

Listing 9. .NET Core route configuration

The back-end followed Code First (23) development when creating the database. This means “Model” was defined first by declaring C# Classes and the database was created accordingly then. Each model, in total of eight, has its own “Controller” to handle a REST call, which means it supports GET, POST, PUT and DELETE. Based on the needs of the data, Controllers were implemented differently and the complexity was various. At the beginning of each controller, the route must be specified and it should be named after the model name for clarification purposes.

After declaring data models and controllers, the next step was to create a SQL Server locally for developing purpose. This could be achieved by using SQL Server (24) provided by Microsoft. By using the code in listing 10, the database and tables will be created automatically based on data models if they do not exist. However, this method just quickly creates required tables and more works need to be done, such as setting nullable and default value of a property.

```
using (var db = new RemoteExpressAnalyticsContext()) {
      db.Database.EnsureCreated();
}
```

Listing 10. Ensure tables are created.

As mentioned in section 4.4, a JavaScript file is called on need-to-be-tracked websites. By default, .NET Core framework denies all external requests to access website contents. Consequently, there is a need to explicitly configure the Cross-Origin Resource Sharing (CORS) as listing 11 to allow external websites to load the deserved file.

```
services.AddCors(options => {
     options.AddPolicy("AllowAllOrigin",
     builder => builder.AllowAnyOrigin().AllowAnyHeader().AllowAnyMethod());
});

app.Map("/analytics.js", map => {
     map.UseCors("AllowAllOrigin ");
});
```

Listing 11. CORS configuration for analytics.js

Microsoft Azure was chosen to host the application and database server. The size of the database and the bandwidth can be modified easily based on the real needs of the project. In addition, for loading page speed, the web server and database server locations should be chosen carefully, which is North Europe in this specific case.

5.2.2 Performance

Whenever the chat application gets new data relating to online users or online customers, the Analytics front-end is notified in real-time. A real-time tracking module was implemented with the help of SignalR library. If users browsers support WebSockets, an advanced web technology, it will be very efficient. Instead of pinging the back-end repeatedly, a bi-directional communication between server and client is established. Consequently, it is only needed to be opened once from the beginning and data can be sent or received between client and server in a nick of time. However, if the client browser does not support WebSockets, SignalR will safely fallback to long polling instead.

5.3 Text analytics implementation

Although this task uses ready-made APIs from IBMWNLU and MATA, the logic to handle the return results need to be carefully considered. It is a must to ensure that all calls to self-implement text analytics API always returns a value, excluding empty-database situation.

5.3.1 IBM Watson Natural Language Understanding Service

IBMWNLU has a SDK for .NET Framework, which means it can be intergrated to an existing product easily. Listing 12 shows parameter configuration before making IBMWNLU API call.

```
public async Task<IHttpActionResult> IBMAnalyze(string content) {
      Parameters parameters = new Parameters() {
      Text = content,
      Features = new Features() {
            Keywords = new KeywordsOptions() {
            Limit = 10,
            Sentiment = true,
            Emotion = true
            }
      }
…
};
```

Listing 12. IBMWNLU parameter configuration

In addition, it is straight forward and easy to configure the returned value after the data is analyzed by IBMWNLU. The language, sentiment, emotion and keywords can be detected by a single API call, which makes IBMWNLU convinient to use. At the time of this writing, IBMWNLU supports detecting keywords for English, French, German, Italian, Portuguese, Russian, Spanish and Swedish. When tested quickly, IBMWNLU worked best with text written in English.

5.3.2 Microsoft Azure Text Analytics Service

On the other hand, MATA is also implemented for study purposes. Because MATA does not provide SDK likes IBMWNLU, it took more time to implement the required methods. Unlike IBMWNLU, MATA is able to perform one propery detection at a time and all other detections except language require text language declaration. It means that in order to extract keyphrases by calling keyphrase API, it is a must to detect the language of the text by calling language API first. At time time of writing, MATA supports detecting keyphrase for English, French, German, Italian, Finnish, Japanese, Polish, Spanish and Swedish. When tested quickly, MATA gave better results in terms of multiple language support. Listing 13 shows a method which was used to detect language by calling MATA API.

```
public async Task<IHttpActionResult> GetLanguage(string description) {
      byteData = Encoding.UTF8.GetBytes(body);
      using (var content = new ByteArrayContent(byteData)) {
            content.Headers.ContentType = new MediaTypeHeaderValue("applica-
            tion/json");
            response = await client.PostAsync(URI_BASE_LANGUAGE, content);
      }
      result = await response.Content.ReadAsStringAsync();
      json = JObject.Parse(result);
…
}
```

Listing 13. Part of a function to retrieve language from MATA

To sum up, each service has its own strength and weakness. The performance and correctness test for both services will be covered in section 6.2

5.4 Time series prediction implementation

A good prediction can only be made from a good dataset. As the company product would be announced publicly in late October, it was difficult to have a qualified dataset. Luckily, some demo days were organized to give customers a general view of the product. Consequently, generated data is ideal for modelling and making prediction. This section used Jupyter Notebook as mentioned in section 2.4

5.4.1 Data preparation

The first step is to prepare a good dataset, which is illustrated in figure 22. All data can be retrieved directly from the Azure database. However, data needs to be filtered because only the one generated in the introduction days is valid. It is not meaningful if data, which was created by developers, is modelled since it does not reflect real life usage. In addition, only a number of chat sessions was modelled for evaluating purpose. It is better to model a specific property of the dataset and optimize the methods for every single attribute than to apply general methods for all attribute. As mentioned in section 4.3.4, this dataset is a time series. Consequently, it is a must to make sure that the data index is a datetime object.

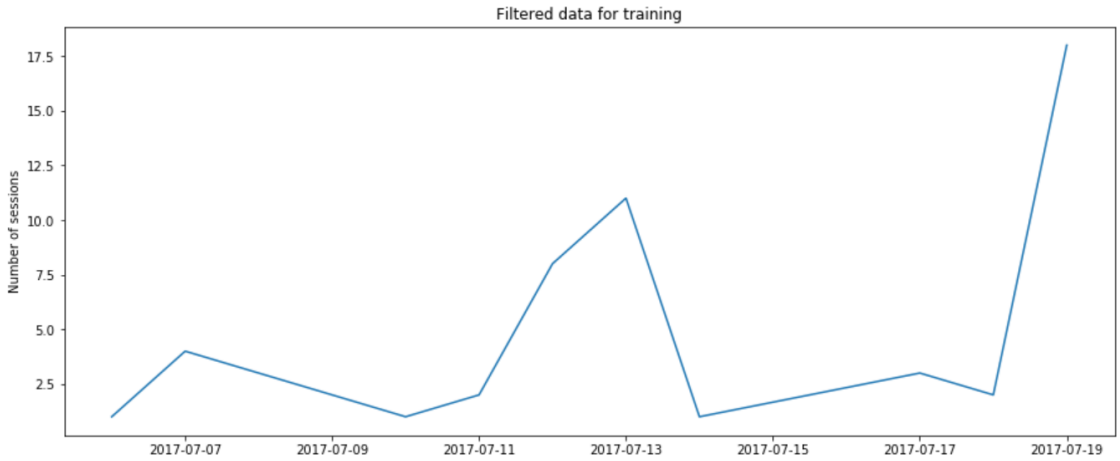

Figure 22. Filtered data for training

Because the dataset was created from July of 2017, it is impossible to collect new data to validate the model. Therefore, the last 10 percent values of the dataset were taken out for model validation, which means the size of dataset to train the model and the size of dataset to validate the model is 10 items and 1 item, respectively.

5.4.2 Data analytics

The second step is to analyze the dataset. Summary statistics of the dataset, which is shown in listing 14, is worth looking into since it gives quick information about the data.

```
count    10.000000
mean      5.100000
std       5.626327
min       1.000000
25%       1.250000
50%       2.500000
75%       7.000000
max      18.000000
```

Listing 14. Summary statistics of sessions number

As shown in figure 22, there is an increasing trend as time passes and the data tends to follow a similar pattern every 4 day. This consumption can be confirmed by a line plot of 4 days grouped data as displayed in figure 23.

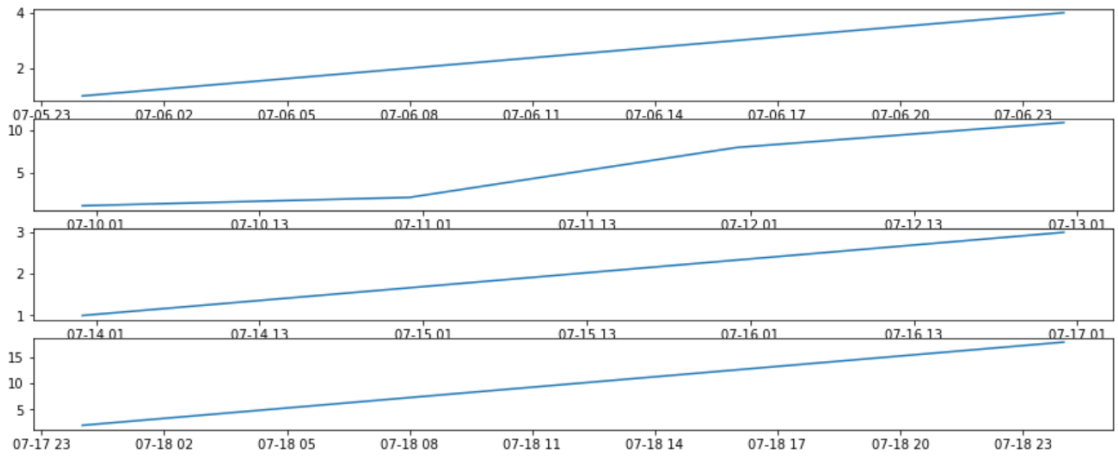

Figure 23. Line plot of 4 days grouped data

As shown in figure 23, it is confirmed that the dataset is not stationary, which means the dataset needs to be applied several transformations to remove trend and pattern.

5.4.3 Data transformation

The third step is to make the dataset become stationary. Because the dataset only contains positive data, log transformation can be applied to eliminate the increasing trend and it is shown in figure 24.

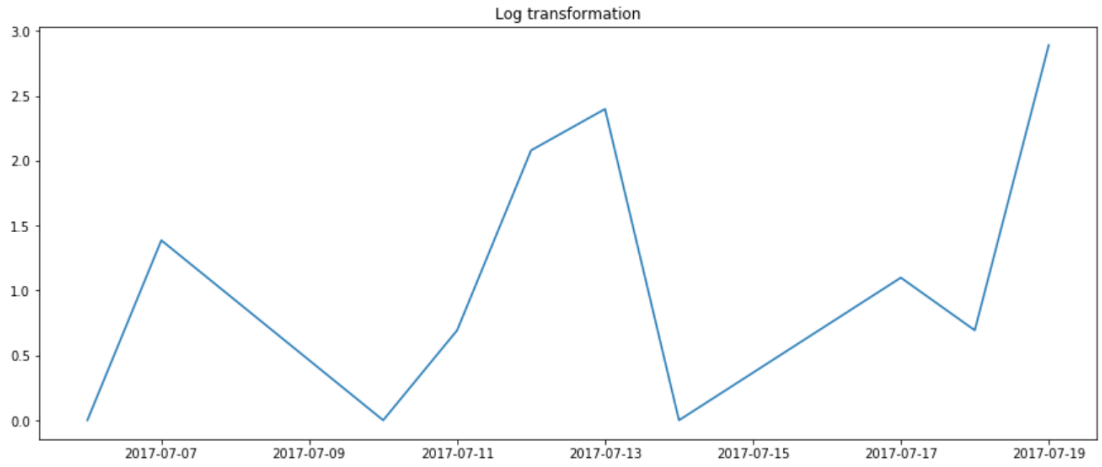

Figure 24. Data set is applied to log transformation

However, the log transformation only reduced the trend and the repeated similar pattern still exists. To make the dataset stationary, first order differencing can be applied as shown in figure 25.

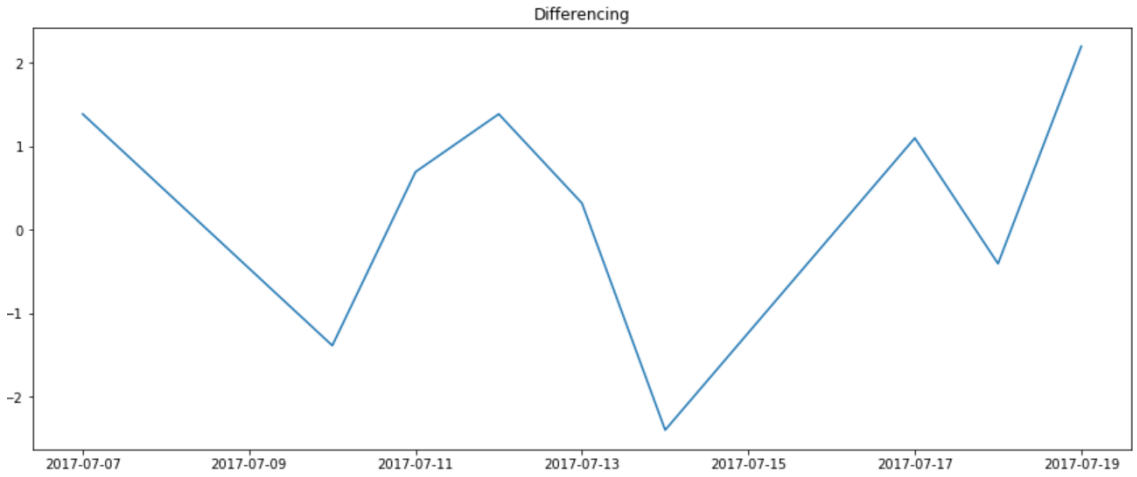

Figure 25. Data set after differencing is applied

After applied log transformation and first order differencing, ADF test can be used to validate whether the dataset is stationary. As shown in listing 15, the p-value is smaller than 0.05 and ADF statistic is smaller than critical value at 5 percent. Therefore, the null hypothesis can be rejected.

```
ADF Statistic                 -3.652342
p-value                        0.004836
#Lags Used                     0.000000
Number of Observations Used    8.000000
Critical Value (1%)           -4.665186
Critical Value (5%)           -3.367187
Critical Value (10%)          -2.802961
```

Listing 15. ADF test result

In addition, ADF test result can be interpreted as with 95 percent of confidence; the dataset is stationary and can be used to build a data model.

5.4.4 Build ARIMA model

In order to build an ARIMA model, 3 parameters p, d and q need to be found. P and q can be determined by Partial Autocorrelation Function(PACF) and Autocorrelation Function (ACF), respectively. According to figure 26, both ACF and PACF show a considerable lag at 1 and 3 and they have the same behavior at some points. Therefore p = 1 and q = 1 is the first guess.

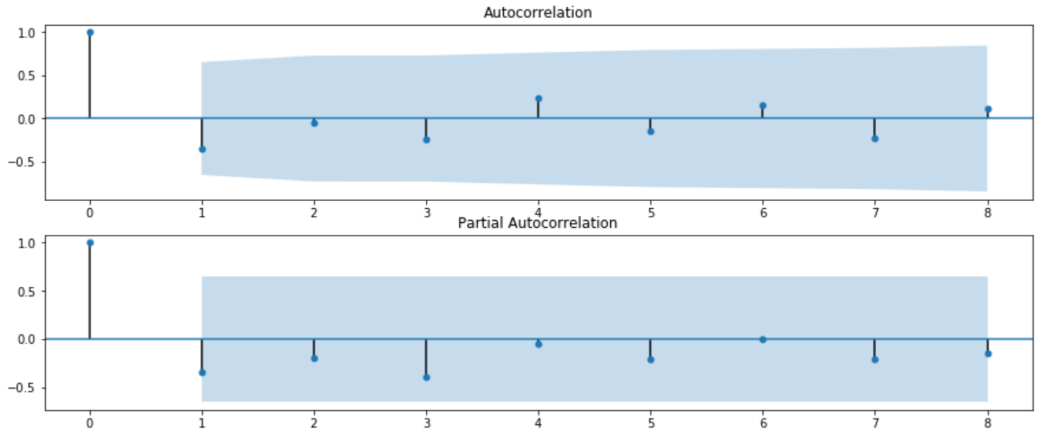

Figure 26. ACF and PACF

Because first order differencing is already applied and ADF test shows that the dataset is stationary, using the new stationary dataset with d = 0 or the old nonstationary dataset with d = 1 will give the same result. Therefore, ARIMA(1,0,1) is a good initial. However, it is better if an automatic script is used to determine the best configuration by running a loop through all possible solutions to get the smallest RMSE. The results from loop test is displayed in listing 16. The pool can be calculated as

$S = P(8,1) * P(3,1) * p(8,1) = 192$ with $p = \{0, \dots, 7\}, d = \{0,1,2\}, q = \{0, \dots, 7\}$

```
ARIMA(0, 0, 1) RMSE=1.459
ARIMA(0, 1, 1) RMSE=1.627
ARIMA(1, 0, 0) RMSE=1.451
ARIMA(1, 1, 0) RMSE=2.181
Best config: ARIMA(1, 0, 0) RMSE=1.451
```

Listing 16. Report from automatic script

As shown in listing 16, it turns out that ARIMA (1,0,0) is the best solution for this dataset since it has the smallest RMSE. All though the iteration is nearly 200, the test only prints 4 ARIMA models, which can be fitted by the given dataset.

5.4.5 Model validation

The final step is to validate the model from section 5.4.4. The summary statistics of predictions as shown in figure 27 suggests that there is a need for bias correction.

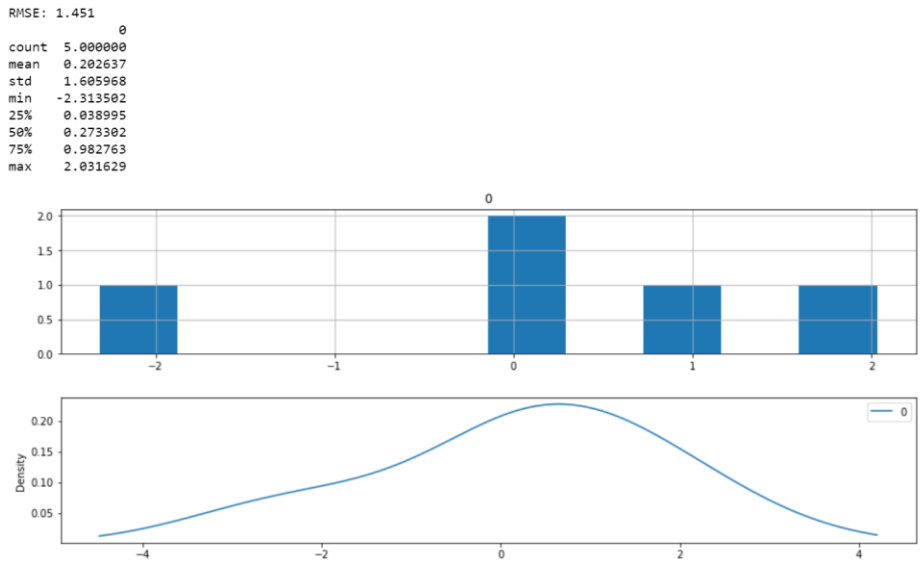

Figure 27. Summary statistic, histogram and density graph of predictions

After having bias correction, the peak of the density plot is slightly shifted to the left as shown in figure 28. In addition, RMSE also improves from 1.451 to 1.436.

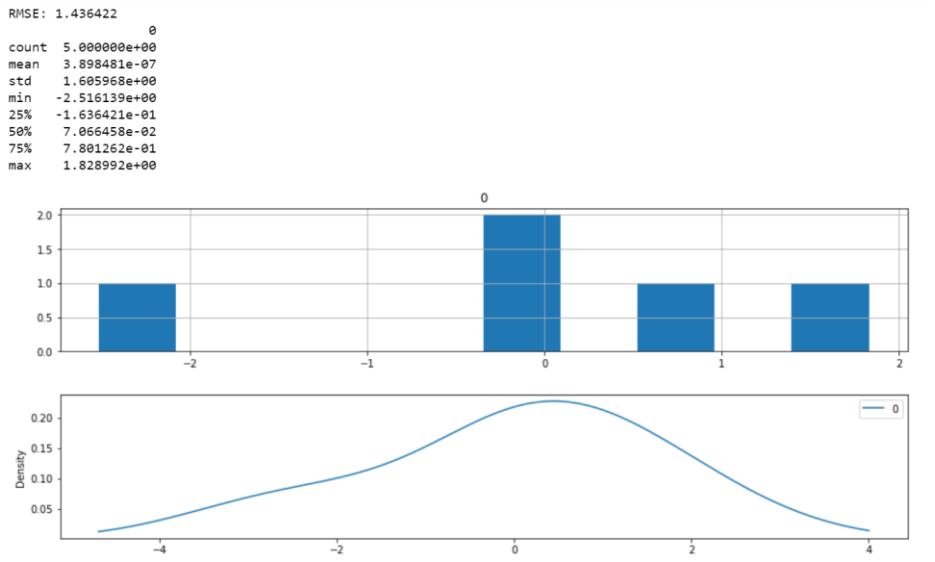

Figure 28. Summary statistic, histogram and density graph of predictions after bias correction

As mentioned in section 4.3.4, the final model needs to pass walk-forward validation and make a prediction for evaluation. The validation now is the last 10 percent data, which has not been used before as mentioned in section 5.4.1, and partly of 90 percent training data. To make it easier to compare the generated data with original data, all predictions were transformed back to the original scale as shown in figure 29.

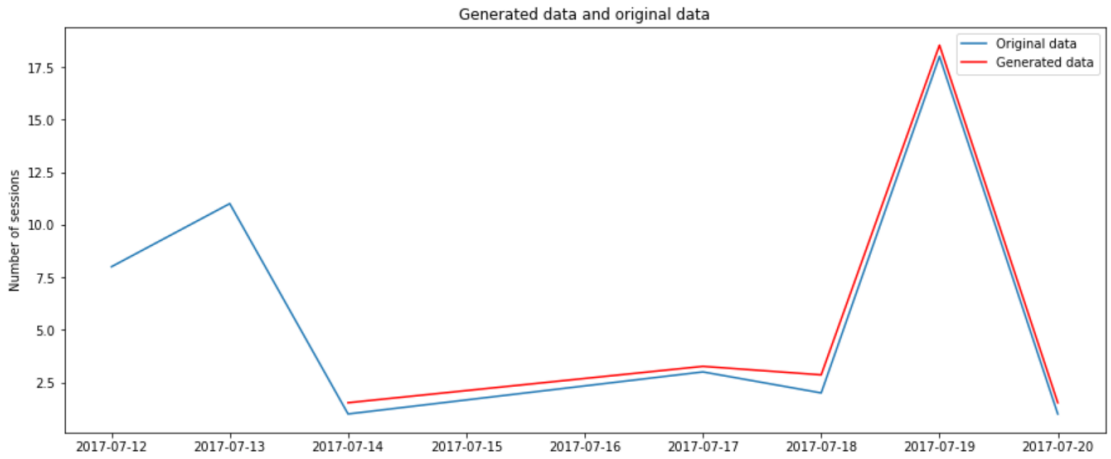

Figure 29. Model fitting

Because the data is transformed before modelling, reverting data back to the original scale needs some trials for the best fit. The prediction from the model is 6.48 sessions

while the actual data is 2 sessions. In addition, other transformation operators were tried to get the data back to the original scale and some of them even gave better results. The most accurate prediction is 2.2 sessions but then the overall fitting would not look as good as in figure 29.

6 Testing

To maintain the stability of the product and to make sure the application behaves as expected, it needs to be tested. As data model testing was mentioned in section 5.4.5, there is no need for further testing relating to it.

6.1 Analytics web application testing

As mentioned in section 2.2, before a code is committed to the master branch, it must pass the testing phase. Protractor, an automation framework, was used for this project. A simple UI test was implemented to check the existing of the element. It was somewhat tricky to set up the test environment for Internet Explorer 11. Part of the configuration to make the test automation works on Internet Explorer 11, Firefox and Chrome are shown in listing 17.

```
multiCapabilities: [{
           'browserName': 'internet explorer',
           'platform': 'ANY',
           'version': '11'
     }{
           'browserName': 'firefox',
           acceptInsecureCerts: true
     },
     {
           'browserName': 'chrome'
     }],
…
localSeleniumStandaloneOpts: {
        jvmArgs: ["-Dwebdriver.ie.driver=IEDriverServer3.4.0.exe",
        "-Dwebdriver.gecko.driver=geckodriver-v0.18.0.exe"]
}
```

Listing 17. Part of Protractor settings

The performance and loading time of the application should also be benchmarked. By using Chrome inspect, they were recorded as shown in figure 30.

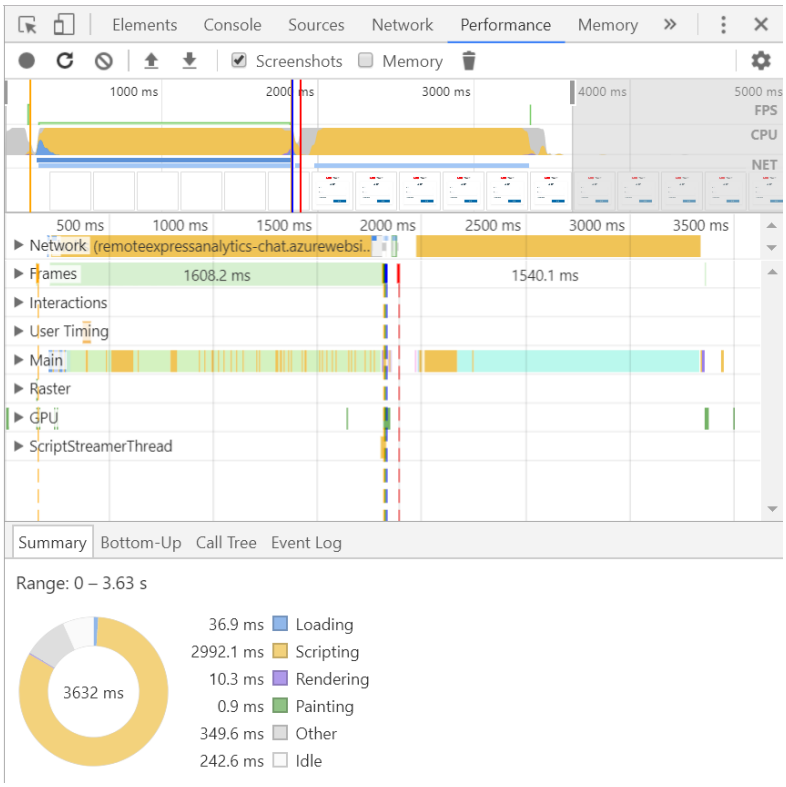

Figure 30. Performance record from Chrome

Because it would take more time to write e2e logic test for test automation, manual testing was preferred. The good point in testing manually is that the performance and the business logic can be checked thoroughly. At the end of this project, all known bugs were fixed and the average loading time of the web application is 2.5 seconds.

6.2 Text analytics testing

In order to test text analytics service, 3 test sets, which were categorized based on text length, were used to check the correctness of results and response time. In addition, each test set included 10 different sentences. To help the reader have a general view, table 2 contains the average of correctness and response time of each test set.

Text length	Properties	MATA	IBMWNLU
Short (<50 char)	Correctness	67%	100%
	Response time	1.031s	0.409s
Medium (50-100 char)	Correctness	100%	100%
	Response time	1.345s	0.559s
Long (>100 char)	Correctness	100%	100%
	Response time	1.377s	0.656s

Table 2. Comparison of MATA and IBMWNLU

As shown in table 2, IBMWNLU is clearly the winner. It has better response times because it requires only 1 API call to detect both language and keywords while MATA requires a separate call for each detection. Furthermore, IBMWNLU also has higher correctness since MATA was unable to detect simple Chinese phrases as good as it should.

7 Operating cost

Besides developing, choosing correct server and its configuration are also important. Selecting the most expensive pricing tier results in wasting of money and using the cheapest one leads to poor performance. In order to select a suitable server, pricing, locations, server technology, traffic and scalability need to be considered. To prevent any problems that could happen, a new chat web app, mobile service and database were created as development environment, which work the same as the existing products. All costs to develop this project are shown in table 3. Table 4 and table 5 display the costs when using MATA and IBMWNLU in production, respectively.

Service name	Price (€/month)
Web Service – Free	0
Mobile Service - Free	0
SQL Server - Basic	4.3
Service Bus - Basic	0.043
Microsoft Azure Text Analytics – Free	0
IBM Watson Natural Language Understanding - Free	0
Sum	4.343

Table 3. Operating cost – Develop environment

Service name	Price (€/month)
Web Service – Standard 1	47.06
Microsoft Azure Text Analytics – Standard 1	210.71
Sum	255.77

Table 4. Operating cost – Production environment with MATA

Service name	Price (€/month)
Web Service – Standard 1	47.06
IBM Watson Natural Language Understanding – Standard 1	45
Sum	92.06

Table 5. Operating cost – Production environment with IBMWNLU

IBM charges by a single API call (25), estimated as total 50,000 calls per month, while Microsoft charges a monthly subscription fee (26). It explains why the estimated cost when using MATA is much higher than IBMWNLU as shown in table 4 and table 5. Because the Analytics application uses the same database with mentioned application, there is no additional cost for SQL Server and Service Bus.

8 Result

8.1 Project outcome

The outcome of this project, fortunately, is quite positive. The Analytics web application was made, which can give the product owner a better view of the products. Illustrations of it can be seen in figure 31, figure 32 and figure 33. There is no limit in the number of registered products and registering a new product to manage is easy. In addition, original products, both mobile and web applications, were also improved to add more functionalities.

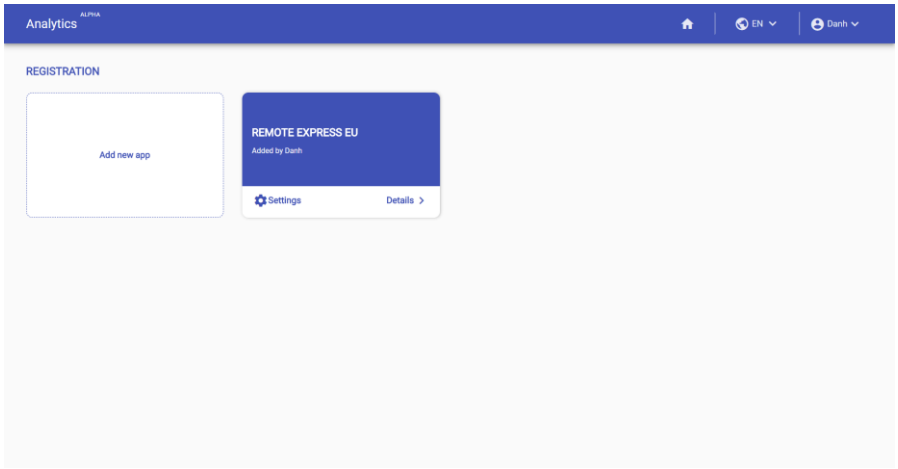

Figure 31. Analytics Web Application – Register View

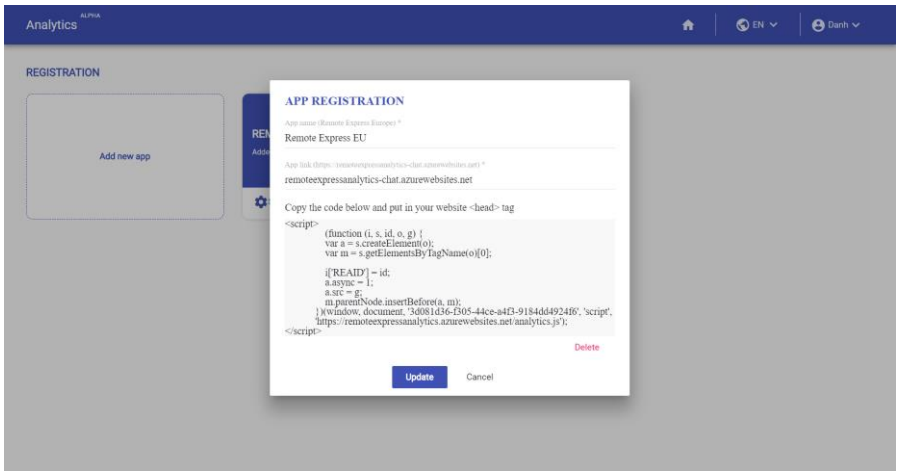

Figure 32. Analytics Web Application – Setting View

As shown in figure 33, The application divided data with different type into different sections for clarification. Users can easily change the time period of the data and choose what kind of data to display. As the layout is responsive, it will resize automatically to make the users feel comfortable. Although this is an analytics application, a module to track data in real-time was implemented, which can help product owners monitor the numbers of online customers and support the users instantly.

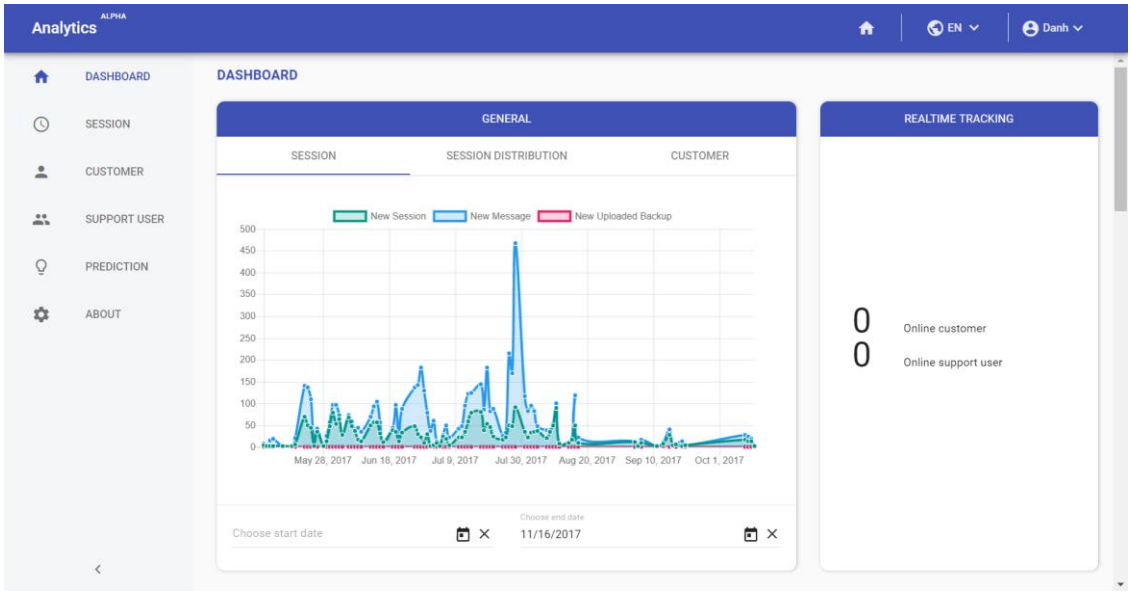

Figure 33. Analytics Web Application – Details View – Dashboard

However, everything has its Achilles' heel. The front-end depends on several dependencies, which need to be updated frequently. Otherwise the application will be outdated within 1 week – or even shorter. This decade has seen the growth of front-end technologies and the following quote is very fitting:

> "You know what. I think we are done here. Actually, I think I'm done. I'm done with the web, I'm done with JavaScript altogether." (27)

Difficulties occurred at some points but they were solved, may be not the best solution but it suits the best to this context. For example, the ARIMA model could give a better prediction but in exchange, the overall fit would not be good anymore. Because this data model is built on a limited dataset and may not be too accurate in a wider time range, more data might give a better result.

8.2 Ability of extension

It is great if a product works with high performance and behaves as it should. It is even better if the product can be extended for further development or integrating with other products. Therefore, ability of extension should be taken into consideration.

8.2.1 Analytics Web Application

On the one hand, by following Don't Repeat Yourself (DRY) principle, all elements were developed in separate module. Consequently, implementing new module and maintenance the project can be done quickly. By sharing database and technologies with the chat application, this project and existing products can be combined easily. It will take approximately 3 days, including testing to deploy the whole project to a new environment. On the other hand, the project is made as general as possible. This means that without a specific data model, the application can smartly display queried data based on data properties. This means with some minor efforts, it can be used to analyze another product.

8.2.2 Text analytics

Even though the outcome from IBMWNLU is as expected, it can be improved by having new customized language models. Theory relating to language models can be found in section 3.1.1. Once the customized language model is trained with Watson Knowledge Studio, it can be deployed to IBMWNLU for evaluating and using. In addition, performance of the trained model can be judged based on accuracy, precision, recall and confusion matrix. An example of confusion matrix can be seen in table 6.

	Class A	Class B	Class C	Class D	Class E
Class A	60		35	5	
Class B	5	70		20	5
Class C	15		80		5
Class D		10		85	5
Class E		5	15	5	75

Table 6. Confusion matrix (28)

Accuracy will measure the correctness, which means it will evaluate the result based on the number of correctly predicted labels. However, a prediction may be predicted with multiple classes. Therefore, precision needs to be considered to measure how many actual classes are correct. Recall will check how well the model remembers the classes of given samples. Confusion matrix is a $N \times N$ matrix, where N is the number of classes. The diagonal indicates correct predictions while others indicate incorrect ones. As in table 6, 60, 70, 80, 85 and 75 are correct predictions while 5, 15, 10 and the rest are incorrect.

8.2.3 Time series prediction

Relating to time series prediction, unfortunately, it cannot be reused since every data has its own pattern and trend. In addition, the data used in this project is inadequate and does not cover all aspects of the chat application. By following the steps written in section 5.4, most of the time series data can be modeled. Based on the complexity of the data, more advanced or more basic methods can be used.

9 Conclusion

"All models are wrong, but some are useful."
-George E. P. Box (29)

Data is the key for everything and smart systems are the future. With enough data, customers can be understood and with smart applications, companies can make their customers happy. The proposal of a customized analytics web application for a specific product was completed and a solution to make the existing app smarter was implemented. However, the actual benefit of this Analytics Web Application depends on how efficiently the team uses the analyzed data.

On the other hand, user privacy and security should be seriously considered. Data is used for analyzing customer behavior and it is not meant to be used to spy the customers. The more data in the database, the more security should be implemented to protect customer data. Relating to this project, only data generated by the chat application product is collected.

Last but not least, analytics is just analytics. A product cannot be improved by just analyzing its data without taking any actions. The ultimate goal is to make customers satisfied, which means interpreted data should be used efficiently to make products better accordingly.

10 References

1. Russell S, Norvig P. Artificial Intelligence - A Morden Approach. 3rd ed. London: Pearson; 2009.
2. Microsoft. What's new in Active Directory Federation Services for Windows Server 2016. [Online]. [cited 2017 November 5]. Available from: https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/overview/whats-new-active-directory-federation-services-windows-server.
3. Microsoft. What is Continuous Integration. [Online]. [cited 2017 November 11]. Available from: https://www.visualstudio.com/learn/what-is-continuous-integration/.
4. Fiorina C. Information: the currency of the digital age. San Francisco: HP; December 6, 2004.
5. IBM. IBM Annual Report. [Online].; 2013 [cited 2017 June 30]. Available from: https://www.ibm.com/annualreport/2013/bin/assets/2013_ibm_annual.pdf.
6. Facebook. Facebook Research. [Online].; 2014 [cited 2017 July 1]. Available from: https://research.fb.com/facebook-s-top-open-data-problems/.
7. YouTube. YouTube for Press. [Online]. [cited 2017 July 1]. Available from: https://www.youtube.com/yt/about/press/.
8. Daymon C, Holloway I. Qualitative Research Methods in Public Relations and Marketing Communications. 2nd ed. London: Routledge; 2010.
9. IBM. IBM SPSS Modeler CRISP-DM Guide. [Online]. [cited 2017 June 17]. Available from: ftp.software.ibm.com/software/analytics/spss/documentation/modeler/14.2/en/CRISP_DM.pdf.
10. Brown MS. Embracing the Data-Mining Process. In Brown MS. Data Mining for Dummies. New Jersey: John Wiley & Sons; 2014. p. 73-88.
11. Matthew N. Data Mining for the Masses: Global Text Project; 2012.
12. IBM. IBM Research. [Online]. [cited 2017 October 18]. Available from: https://www.research.ibm.com/deepqa/faq.shtml.
13. Schneider C. IBM - The biggest data challenges that you might not even know you have. [Online].; 2016 [cited 2017 October 19]. Available from: https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know/.

14. Google. Google Books Ngram Viewer. [Online]. [cited 2017 October 19]. Available from: https://books.google.com/ngrams/.
15. Hearst MA. Automatic Acquisition of Hyponyms from Large Text Corpora. In COLING ; 1992; Nantes France. p. 3.
16. Brockwell PJ, Davis RA. Introduction to Time Series and Forecasting. 2nd ed. New York: Springer; 2002.
17. DataMarket. Monthly number of employed persons in Australia. [Online]. [cited 2017 September 1]. Available from: https://datamarket.com/data/set/22t8/monthly-number-of-employed-persons-in-australia-thousands-feb-1978-apr-1991.
18. World Bank Group. The World Bank Group. [Online]. [cited 2017 September 1]. Available from: https://data.worldbank.org/indicator/SP.POP.TOTL.
19. MathWorks. Moving Average Model. [Online]. [cited 2017 October 1]. Available from: https://se.mathworks.com/help/econ/moving-average-model.html.
20. John W. Your Brand's True Colours. [Online].; 2007 [cited 2017 July 18]. Available from: https://www.entrepreneur.com/article/175428.
21. Microsoft. sys.syslanguages (Transact-SQL). [Online]. [cited 2017 September 5]. Available from: https://docs.microsoft.com/en-us/sql/relational-databases/system-compatibility-views/sys-syslanguages-transact-sql.
22. Microsoft. HttpStatusCode Enumeration. [Online]. [cited 2017 August 29]. Available from: https://msdn.microsoft.com/en-us/library/system.net.httpstatuscode.aspx.
23. Microsoft. Entity Framework Code First to a New Database. [Online]. [cited 2017 August 30]. Available from: https://msdn.microsoft.com/en-us/library/jj193542%28v=vs.113%29.aspx?f=255&MSPPError=-2147217396.
24. Microsoft. SQL Server 2016. [Online]. [cited 2017 September 5]. Available from: https://www.microsoft.com/en-us/sql-server/sql-server-2016.
25. IBM. Natural Language Understanding. [Online]. [cited 2017 October 4]. Available from: https://www.ibm.com/watson/services/natural-language-understanding/.
26. Microsoft. Cognitive Services Pricing. [Online]. [cited 2017 October 4]. Available from: https://azure.microsoft.com/en-us/pricing/details/cognitive-services/text-analytics/.
27. Aguinaga J. Hackernoon. [Online].; 2016 [cited 2017 October 5]. Available from: https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f.

28. IBM. Train and evaluate custom machine learning models of Watson Developer Cloud. [Online].; 2017 [cited 2017 October 20]. Available from: https://developer.ibm.com/dwblog/2017/machine-learning-custom-models-watson-developer-cloud/.
29. Box GEP, Hunter W, Hunter JS. Statistics for Experimenters. 2nd ed.: Wiley-Interscience; 2005.

Survey Question

This survey includes 9 questions divided into 2 parts and it will take around 15 minutes to complete.

Your responses to this survey will be kept anonymous.

Thank you for your participation.

Part 1: In the first part of the survey, you will find 6 questions relating to analytics tools.

1. Have you ever used / going to use any analytics services? (Check all that apply)
2. How do those analytics services help you to manage your applications/ websites?
3. How long have you been using those analytics services?
4. How often do you check those analytics services?
5. How long does each session take?
6. What do you want those services to have / to improve?

Part 2: This section contains the last three questions relating to Machine Learning.

1. Do you know about Machine Learning?
2. What do you think about an application with Machine Learning implementation?
3. Finally, do you think it is necessary for a product to have Machine Learning Service?

Comparison of Microsoft Cognitive API and IBM Watson

A return result from Microsoft Cognitive API includes key phrases and sentiment level (from 0%, very negative – 100%, very positive). On the other hand, IBM Watson returns sentiment level (0 to +1 for positive meaning and 0 to -1 for negative one). In addition, it also provides the relevance level (from 0 to 1) of a keyword on given text.

For the sake of clarity, sentiment level and relevance level from IBM Watson results will be converted to percentage by following formulars:

$$f(IBM_{relevance}) = IBM_{relevance} * 100$$
$$g(IBM_{sentiment}) = (IBM_{sentiment} * 100 + 100\ )\ /2$$

Complexity & Sentiment		**Sentence**	**Microsoft Cognitive API**	**IBM Watson API**
Simple	Positive	Your suggestion is very helpful	Suggestion	Suggestion (99%)
			98%	89%
		Thank you for your help	Help	Help (99%)
			100%	94%
		The drive works perfectly now	Drive works	Drive (90%)
			87%	91%
		I had a wonderful experience!	Wonderful experience	Wonderful experience (93%)
			98%	94%

		Your reply was fast and helpful	Reply	Reply (96%)
			98%	87.5%
	Negative	Your suggestion does not work	Suggestion	Suggestion (96%)
			50%	14%
		I do not like this service	Service	Service (91%)
			15%	17%
		My ACS580 does not work	ACS580	ACS580 (90%)
			19%	10.5%
		I had a terrible experience	Terrible experience	Terrible experience (95%)
			2%	6.5%
		Backup feature does not	Backup feature	Backup feature (96%)
			50%	17%
Complex	Positive	I had a wonderful experience. The answer from support service was helpful.	Answer, support service, wonderful experience	Wonderful experience (98%), support service (86%), answer (74%)
			100%	93.5%
		The solution is straight forward and the drive works perfectly again	Drive, solution	Solution (98%), drive (94%)
			100%	95.5%
		I like the backup feature since it helps me share the backup files with my colleagues	Backup files, backup feature, colleagues	Backup feature (97%), colleagues (69%)
			99%	90%

		The Drivetune app not only allows me to control ACS580 remotely but also has a better design in comparison to the panel	Better design, comparison, panel, Drivetune app	Drivetune app (91%), better design (75%), comparison (44%), panel (44%)
			85%	50%
		The newest firmware fixed a lot of problems on my drive and now the drive is working as it should. Great work.	Drive, great work, lot of problems, newest firmware	Newest firmware (96%), great work (77%), drive (51%), problems (48%)
			24%	61.5%
	Negative	The backup feature has caused a lot of problems. The home menu disappeared after performing backup restore	Backup feature, home menu, lot of problems, performing backup	Backup restore (100%), backup feature (95%), home menu (80%), problems (47%)
			3%	24%
		The configuration you sent does not work with my Drive. The panel said "please config the voltage"	Configuration, drive, panel, voltage	Config (99%), configuration (91%), voltage, (88%), panel (81%), drive (57%)
			50%	24.5%
		After restoring the backup file sent from your service, my drive cannot be connected anymore. The "restore to default" feature does not help also. The drive is ACS580 and firmware version is ACCN12.12.123	ACCN12, ACS580, drive, feature, firmware version, service	Backup file (95%), firmware version (87%), drive (65%), service (50%), feature(49%), ACS580 (48%)
			83%	50%

		The response from this service is not helpful at all and I think I have been wasting time and money for this.	Money, response, service, time	Response (92%), service (91%), time (87%), money (87%)
			14%	10%
		Backup feature does not on ACS380 because when I tried to click “backup restore”, the app crashed.	ACS380, app , backup feature	backup restore (98%), backup feature (91%), app (77%), ACS380 (62%)
			50%	22%
Nonsense		Tehokas kone but I haves problmes with it immediately	haves problmes, Tehokas kone	Tehokas kone (94%), problmes (68%)
			50%	50%
		Do your knew the date I ACS580 firmware upgrade ?	ACS580 firmware, date	firmware upgrade (95%), date (54%)
			50%	50%