



Project Title	Maersk Line Container Management System (CMS)
Module Code	CT071-5-3-DDAC
Intake Code	UC3F1706SE
Student ID	TP035869
Student Name	MAGGIE YONG MEI CHI
Lecturer Name	DR. KALAI ANAND A/L RATNAM
Hand in Date	13th APRIL 2018

ACKNOWLEDGEMENT

The Designing and Developing of Application on the Cloud (DDAC) assignment gave me a large opportunity and great chance for learning and professional full development of application. I am grateful for having a chance to meet many wonderful professionals who led me through this assignment.

I am using this opportunity to express my deepest gratitude and special thanks to Dr. Kalai Anand A/L Kunasekaran, my DDAC lecturer who despite of being extraordinarily busy with his duties, took some time to hear and guide me. Also, I appreciate for his carefulness and precious guidance, which were extremely valuable and important for my study in both theoretically and practically. When I am facing problems in my DDAC assignment, I will email him regarding the problem that I faced. He guided me patiently, step-by-step to make sure I am on the right track.

I would like to express my deepest thanks to my classmates, who helped me whenever I faced any problem. They gave me useful decision and necessary advices on the problems that I faced. We discussed together as well due to some of the problems we faced might be similar to each other.

Finally, I wish to thank my parents for their support and encouragement throughout my DDAC assignment. They aided me whenever I am stuck in problems and encouraged me throughout the thick and thin.

Table of Contents

Chapter 1: INTRODUCTION.....	6
1.1 Project Background.....	6
1.2 Project Objective.....	7
1.3 Project Scope.....	7
1.4 Requirement Specification.....	8
1.5 Summary of Major Functions	9
Chapter 2: PROJECT PLAN	10
2.1 Work Breakdown Structure (WBS)	10
2.2 Gantt Chart.....	12
Chapter 3: DESIGN	14
3.1 Cloud Architecture	14
3.2 Design Consideration.....	15
3.3 Modelling	16
3.3.1 Use Case Diagram.....	16
3.3.2 Use Case Description.....	17
3.3.3 Sequence Diagram	24
3.3.4 Class Diagram.....	29
CHAPTER 4: IMPLEMENTATION	30
4.1 ASP.NET Web Application	30
4.2 Azure Resource Group.....	31
4.2.1 Resource Group of Web Application Service.....	31
4.2.2 Resource Group of SQL Database.....	32
4.2.3 Resource Group of Traffic Manager.....	33
4.3 Azure SQL Server	34

4.3.1	Setup SQL Servers	34
4.4	Azure SQL Database	36
4.4.1	Setup Primary SQL Database	36
4.4.2	Setup Secondary SQL Database	37
4.5	Azure Web Application Service.....	41
4.5.1	Setup Web Application Service	41
4.5.2	Deploy APS.NET Web Application	43
4.6	Azure Traffic Manager	47
4.6.1	Setup Traffic Manager Profile	47
4.6.2	Setup Endpoints	48
4.6.3	Testing Traffic Manager Profile	49
4.7	Azure Web Application Autoscaling	50
4.7.1	Setup Web Application Scale Out	50
4.7.2	Setup Scale Rule for Auto Scaling Plan	51
4.8	System Interface	52
4.8.1	Login	52
4.8.2	Main Page	53
4.8.3	Register Agent	54
4.8.4	Manage Customer	55
4.8.5	Manage Booking	57
4.8.6	Manage Schedule	58
4.8.7	Manage Ship	60
Chapter 5: TESTING		61
5.1	Performance Testing	61
5.2	Unit Testing.....	63

5.2.1	Login	63
5.2.2	Register Agent	64
5.2.3	Manage Customer	65
5.2.4	Manage Booking	66
5.2.5	Manage Schedule	67
5.2.6	Manage Ship	68
Chapter 6: CONCLUSION		69
Chapter 7: REFERENCE		70
Chapter 8: APPENDIX		71

Chapter 1: INTRODUCTION

1.1 Project Background

Maersk Line is the global container division and the largest operating unit of the A.P. Moller – Maersk Group, a Danish business conglomerate. It is the world's largest container shipping company having customers through 374 offices in 116 countries. It employs approximately 7,000 sea farers and approximately 25,000 land-based people. Maersk Line operates over 600 vessels and has a capacity of 2.6 million TEU. The company was founded in 1928.

Operating in 100 countries and transporting goods around the globe, at first glance it would appear Danish shipping company Maersk Line is already handling all the cargo it can manage. But when Maersk determined that the volume of most of the goods it was shipping had grown to full capacity, the company decided that cloud powered solutions would be a crucial part of rectifying the situation.

“There was a ‘mind-opener’ where Maersk said, ‘How can we support the overall business strategy, and also from an IT perspective,’” says Soeren Lorenzen, an account general manager with Hewlett-Packard company who is involved first-hand with Maersk’s ITO efforts. “There was a new CIO who wanted to outsource every part of IT, but without [negatively] impacting shipping.”

In an effort to support further business growth and increase organizational flexibility, Maersk decided to consolidate all of its data centres and server rooms operating worldwide onto a virtualized platform. Microsoft Azure was already hosting some of Maersk’s IT environment, and in March 2016 Maersk initially approached Microsoft about expanding the scope of the relationship. Moving forward, Lorenzen says Maersk is currently changing over its IT setup based on Microsoft Azure, starting with the desktop environment up to container management.

1.2 Project Objective

The project's objectives are listed below:

- To develop a Container Management System (CMS) application that developed using Microsoft Azure Cloud Service.
- To upgrade and maintain the performance of Container Management System (CMS) application after the system go-live.
- To design Container Management System (CMS) application that able to meet the business specifications or user requirements.
- To develop a Container Management System (CMS) application that compatible with most web browser in the market.

1.3 Project Scope

The project scope for Container Management System (CMS) are:

- Able to manage the whole booking process which is from searching schedule to booking confirmation.
- Able to manage cost such as increase profit and productivity, reduce errors, overheads and recurring costs, and optimise resources.
- Able to access Container Management System (CMS) application from different geographical areas.
- Able to customise the solution of business specifications during peak seasons.

1.4 Requirement Specification

In Container Management System (CMS) application, there are several requirement specifications are listed as followed:

1. Availability

The Container Management System (CMS) application must be accessible on the online web most of the time.

2. Scalability

The Container Management System (CMS) application must be scalable enough to meet business specifications and users' requirements.

3. Maintainability

The Container Management System (CMS) application must be able to upgrade and maintain frequently. Meanwhile, users able to use the system simultaneously.

4. Monitoring

The Container Management System (CMS) application must be monitored most of the time to identify and determine the errors or defects of the system and fix them as soon as possible.

5. Provisioning

The new Container Management System (CMS) application must be able to provision to the Microsoft Azure Platform.

1.5 Summary of Major Functions

In Container Management System (CMS) application, the summary of the major functions, which also considered as functional requirement specification, is listed as followed:

1. Admin

- Login
- Register new agent
- Create, Read, Update and Delete schedule
- Create, Read, Update and Delete ship
- Create, Read, Update and Delete customer
- Create and Read booking

2. Agent

- Login
- Create, Read, Update and Delete customer
- Create and Read booking

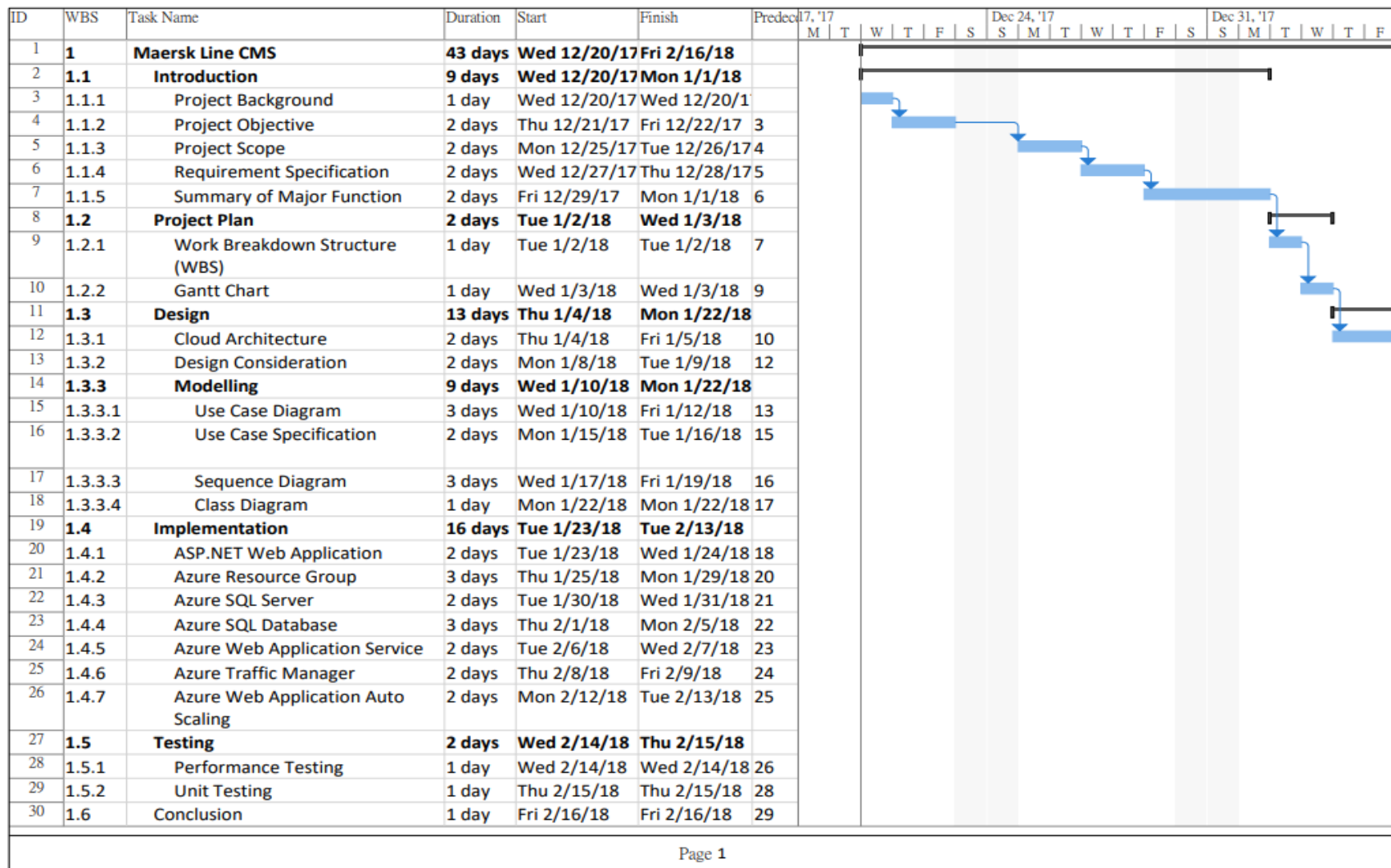
Chapter 2: PROJECT PLAN

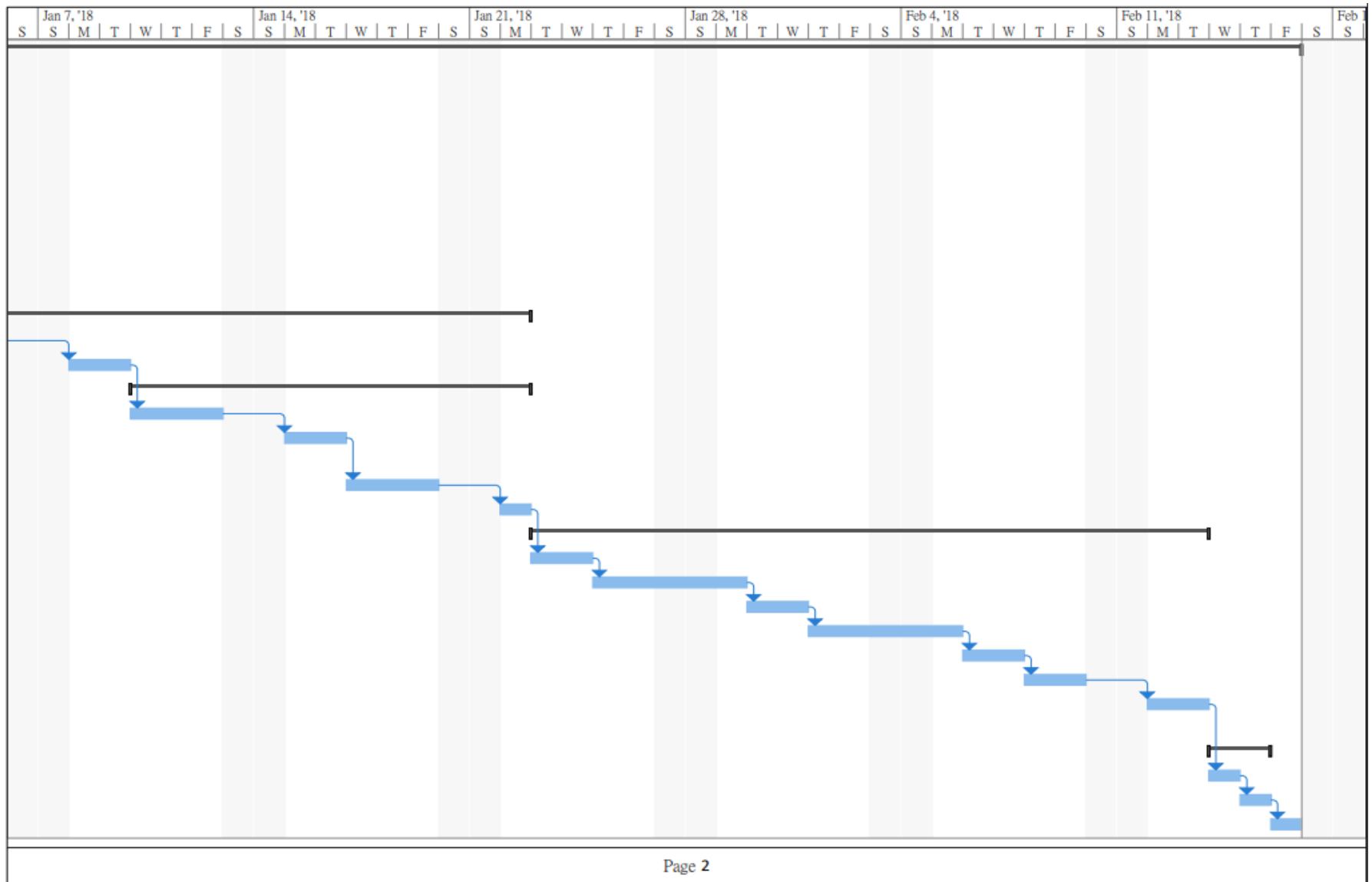
2.1 Work Breakdown Structure (WBS)

WBS	Task Name	Duration
1	Maersk Line CMS	43 days
1.1	Introduction	9 days
1.1.1	Project Background	1 day
1.1.2	Project Objective	2 days
1.1.3	Project Scope	2 days
1.1.4	Requirement Specification	2 days
1.1.5	Summary of Major Function	2 days
1.2	Project Plan	2 days
1.2.1	Work Breakdown Structure (WBS)	1 day
1.2.2	Gantt Chart	1 day
1.3	Design	13 days
1.3.1	Cloud Architecture	2 days
1.3.2	Design Consideration	2 days
1.3.3	Modelling	9 days
1.3.3.1	Use Case Diagram	3 days
1.3.3.2	Use Case Specification	2 days
1.3.3.3	Sequence Diagram	3 days
1.3.3.4	Class Diagram	1 day
1.4	Implementation	16 days
1.4.1	ASP.NET Web Application	2 days
1.4.2	Azure Resource Group	3 days
1.4.3	Azure SQL Server	2 days
1.4.4	Azure SQL Database	3 days
1.4.5	Azure Web Application Service	2 days
1.4.6	Azure Traffic Manager	2 days
1.4.7	Azure Web Application Auto Scaling	2 days
1.5	Testing	2 days

1.5.1	Performance Testing	1 day
1.5.2	Unit Testing	1 day
1.6	Conclusion	1 day

2.2 Gantt Chart





Chapter 3: DESIGN

3.1 Cloud Architecture

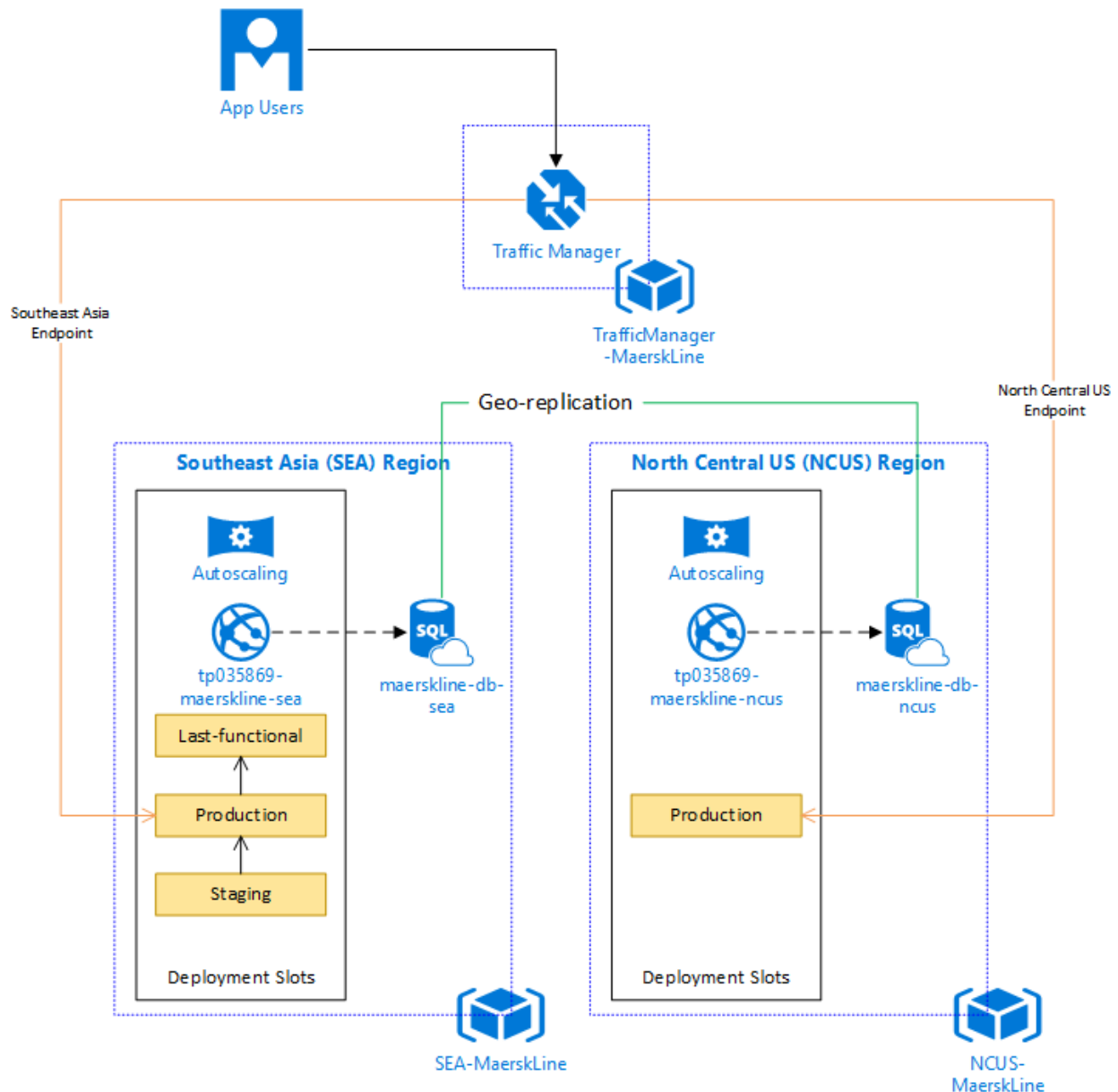


Figure 1 Cloud Architecture of Maersk Line

Figure 1 shows the overall of cloud architecture that implemented to deploy Maersk Line Web Application to Microsoft Azure Cloud Platform. This cloud platform allows individual to view or use the web application in different regions around the globe. For Maersk Line, the primary cloud service is in Southeast Asia and secondary service is in North Central United States. These 2

regions can have up to 3 autoscaling service and each of these has their own database. This is to allow the performance of the database can be fast because the location of databases and application are much nearer. Traffic Manager will manage the performance of the web application by introducing the endpoints for both regions into the service. The geo-replication is used in order to ensure when the primary database (SEA) is down, the secondary will take place and continue its work. This allows to improve the availability of Maersk Line Container Management System (CMS).

3.2 Design Consideration

The designs that needed to be considered before entering to the design phase are listed as followed:

1. The Container Management System (CMS) should be able to reduce all the redundant costs such as recurring cost, supply chain cost or overheads.
2. The Container Management System (CMS) should be able to provide an effective and efficient system for customer to manage the business process by cloud-based application.
3. The Container Management System (CMS) should be able to meet customers' satisfaction.

However, there are several things needed to be taken into account. A limited Azure budget was given to the development team which insufficient to develop the project. Besides, time constraint is another factor to develop the system. Hence, the team decided to host the system as a cloud-based system on-line so given Azure budget will not be exceeded.

3.3 Modelling

3.3.1 Use Case Diagram

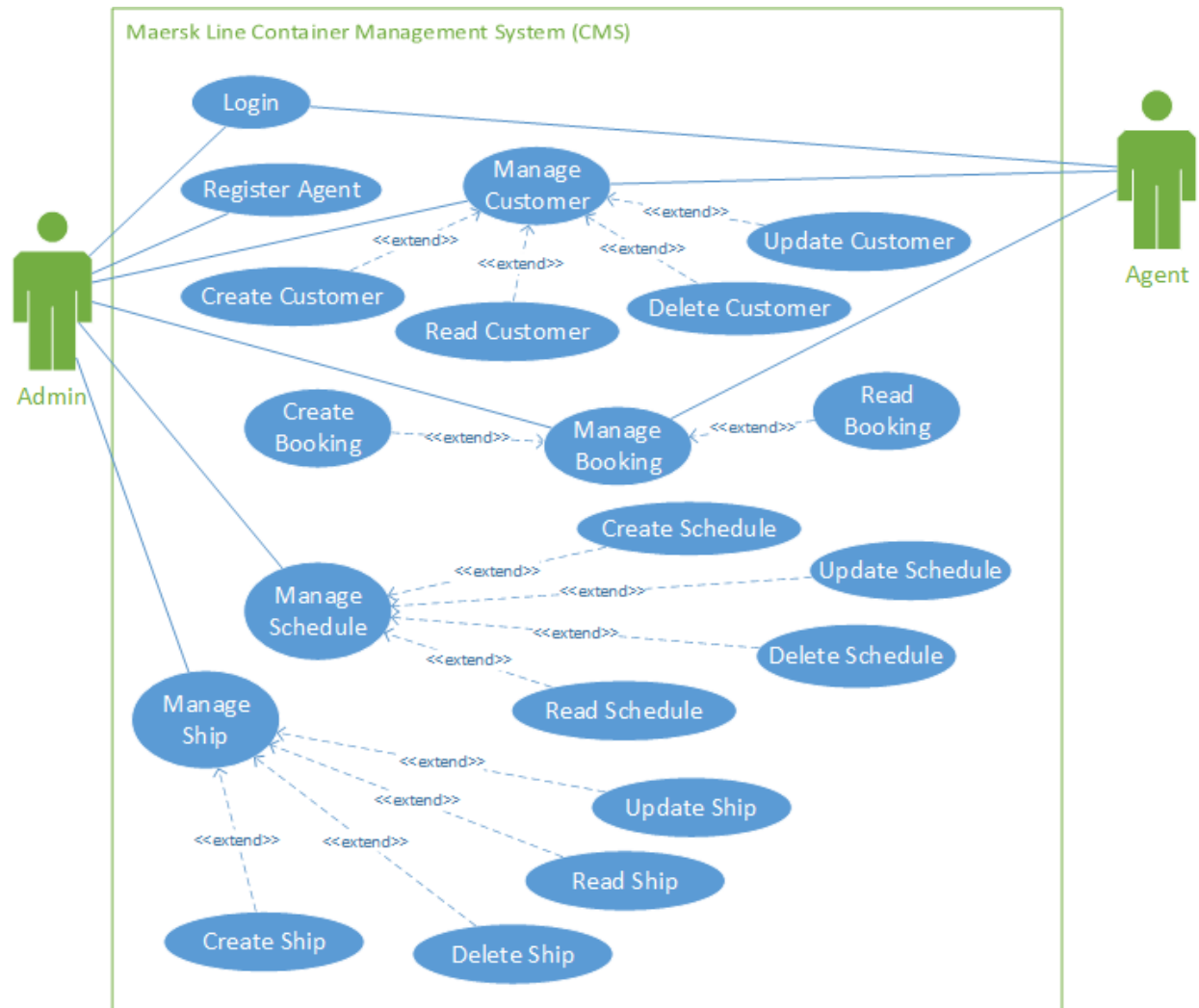


Figure 2 Use Case Diagram

3.3.2 Use Case Description

Use Case	Login
Summary	Users enter credentials to login to the system.
Dependency	-
Actor	Admin, Agent
Precondition	-
Description of Main Sequence	<ol style="list-style-type: none">1. Enter email and password.2. The email and password entered will be verified by system.3. Users will be directed to their home page respectively.
Alternative Flow	2(a). Error message will be shown if users enter the wrong credentials.
Post Condition	Admin and agent logged into the system successfully.

Use Case	Register Agent
Summary	Admin enters agents' information to register new agent.
Dependency	-
Actor	Admin
Precondition	Admin must log in to the system.
Description of Main Sequence	<ol style="list-style-type: none">1. Admin enters information of the new agent.2. The information entered will be verified by system.3. A message will be shown to admin that indicates agent registered successfully.

Alternative Flow	2(a). Error message will be shown if admin enter the wrong information.
Post Condition	Agent account created successfully.

Use Case	Manage Customer
Summary	Users are able to create, read, update and delete customers' records.
Dependency	<<extend>> Create Customer <<extend>> Read Customer <<extend>> Update Customer <<extend>> Delete Customer
Actor	Admin, Agent
Precondition	Admin and agent must log in to the system.
Description of Main Sequence	<u>Create Customer</u> 1. Click on "Create new customer". 2. Enter new customers' information. 3. The new customers' information entered will be verified by system. 4. New customer created successfully. <u>Read Customer</u> 1. Select the particular record of the customer and click "View". 2. Customer's record is displayed successfully. <u>Update Customer</u> 1. Select the particular record of the customer and click "Update".

	<ol style="list-style-type: none"> 2. Enter updated information of the customer. 3. The new customers' information entered will be verified by system. 4. Record of customer is updated successfully. <p><u>Delete Customer</u></p> <ol style="list-style-type: none"> 1. Select the particular record of the customer and click "Delete". 2. Customer's record is deleted successfully.
Alternative Flow	<p><u>Create Customer</u></p> <p>3(a). Error message will be shown if users enter the wrong information.</p> <p><u>Update Customer</u></p> <p>3(a). Error message will be shown if users enter the wrong information.</p>
Post Condition	-

Use Case	Manage Booking
Summary	Users are able to create and read customers' bookings.
Dependency	<<extend>> Create Booking <<extend>> Read Booking
Actor	Admin, Agent
Precondition	Admin and agent must log in to the system.
Description of Main Sequence	<p><u>Create Booking</u></p> <ol style="list-style-type: none"> 1. Click on "Create new booking". 2. Enter new bookings' information.

	<ol style="list-style-type: none">3. The new bookings' information entered will be verified by system.4. New booking created successfully. <p><u>Read Booking</u></p> <ol style="list-style-type: none">1. Select the particular record of the booking and click "View".2. Bookings' record displayed successfully.
Alternative Flow	<p><u>Create Booking</u></p> <p>3(a). Error message will be shown if users enter the wrong information.</p>
Post Condition	-

Use Case	Manage Schedule
Summary	Users are able to create, read, update and delete schedule records.
Dependency	<<extend>> Create Schedule <<extend>> Read Schedule <<extend>> Update Schedule <<extend>> Delete Schedule
Actor	Admin
Precondition	Users must log in to the system.
Description of Main Sequence	<p><u>Create Schedule</u></p> <ol style="list-style-type: none">1. Click on "Create new schedule".2. Enter new schedule information.3. The new schedule information entered will be verified by system.

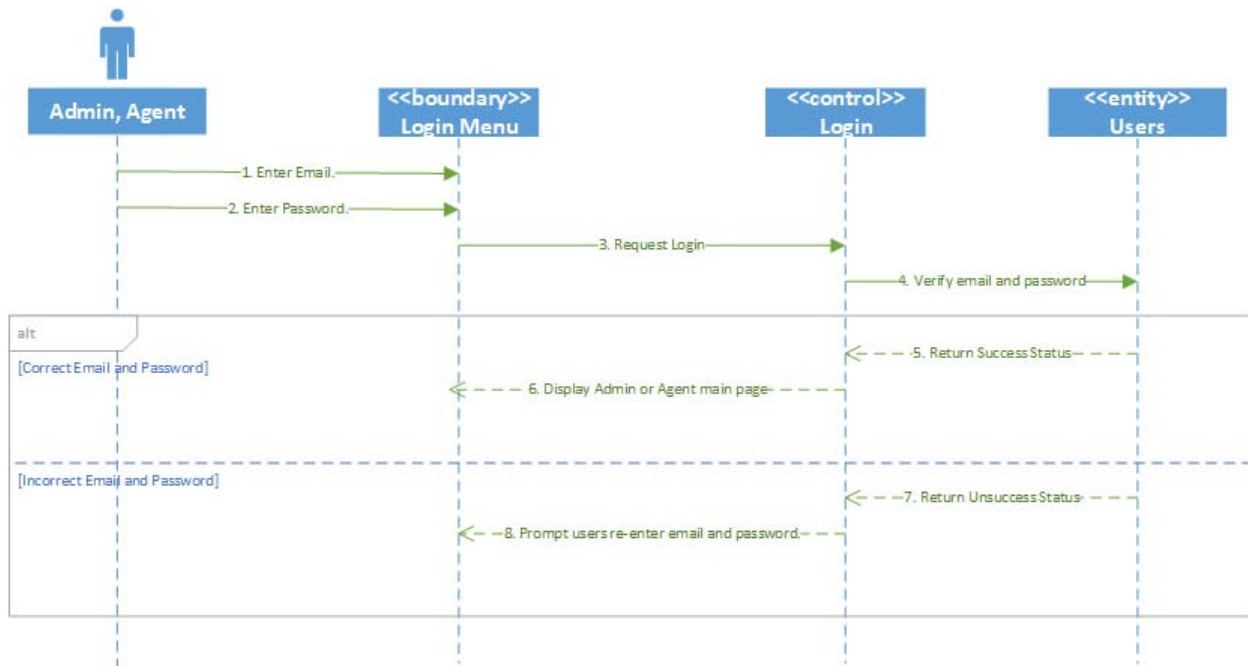
	<p>4. New schedule record created successfully.</p> <p><u>Read Schedule</u></p> <p>1. Select the particular record of the schedule and click “View”.</p> <p>2. Schedule record is displayed successfully.</p> <p><u>Update Schedule</u></p> <p>1. Select the particular record of the schedule and click “Update”.</p> <p>2. Enter updated information of the schedule.</p> <p>3. The new schedule information entered will be verified by system.</p> <p>4. Record of schedule is updated successfully.</p> <p><u>Delete Schedule</u></p> <p>1. Select the particular record of the schedule and click “Delete”.</p> <p>2. Schedule record is deleted successfully.</p>
Alternative Flow	<p><u>Create Schedule</u></p> <p>3(a). Error message will be shown if users enter the wrong information.</p> <p><u>Update Schedule</u></p> <p>3(a). Error message will be shown if users enter the wrong information.</p>
Post Condition	-

Use Case	Manage Ship
Summary	Users are able to create, read, update and delete ship records.
Dependency	<<extend>> Create Ship <<extend>> Read Ship <<extend>> Update Ship <<extend>> Delete Ship
Actor	Admin
Precondition	Users must log in to the system.
Description of Main Sequence	<u>Create Ship</u> <ol style="list-style-type: none">1. Click on “Create new ship”.2. Enter new ship information.3. The new ship information entered will be verified by system.4. New ship record created successfully. <u>Read Ship</u> <ol style="list-style-type: none">1. Select the particular record of the ship and click “View”.2. Ship record is displayed successfully. <u>Update Ship</u> <ol style="list-style-type: none">1. Select the particular record of the ship and click “Update”.2. Enter updated information of the ship.3. The new ship information entered will be verified by system.4. Record of ship is updated successfully.

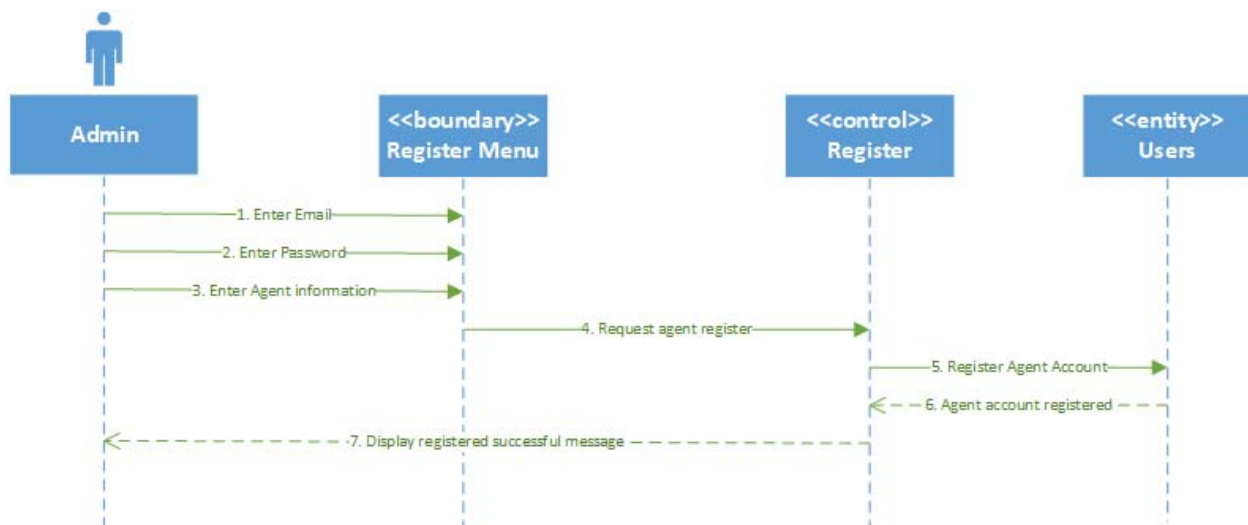
	<u>Delete Ship</u> 1. Select the particular record of the ship and click “Delete”. 2. Ship record is deleted successfully.
Alternative Flow	<u>Create Ship</u> 3(a). Error message will be shown if users enter the wrong information. <u>Update Ship</u> 3(a). Error message will be shown if users enter the wrong information.
Post Condition	-

3.3.3 Sequence Diagram

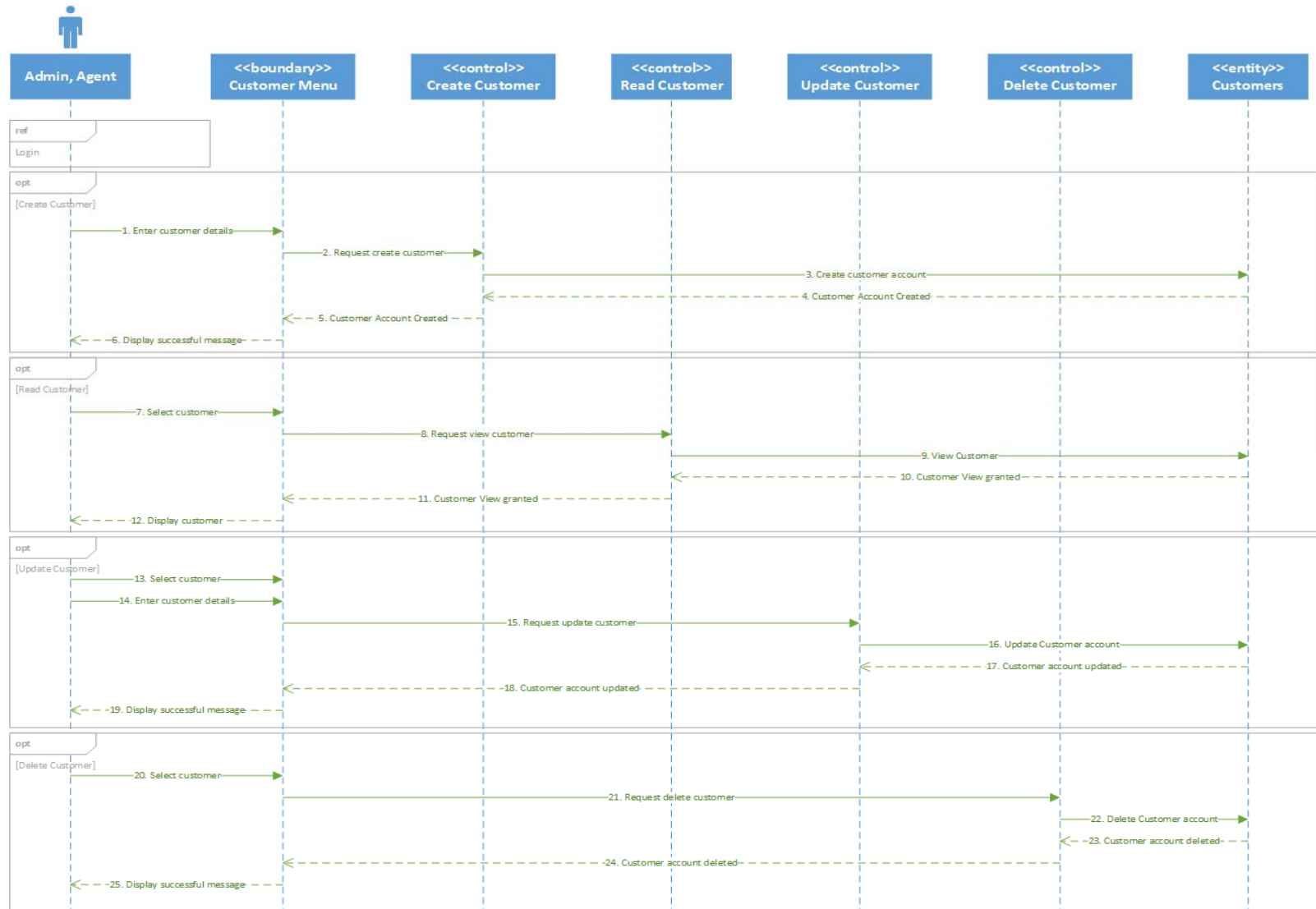
3.3.3.1 Login



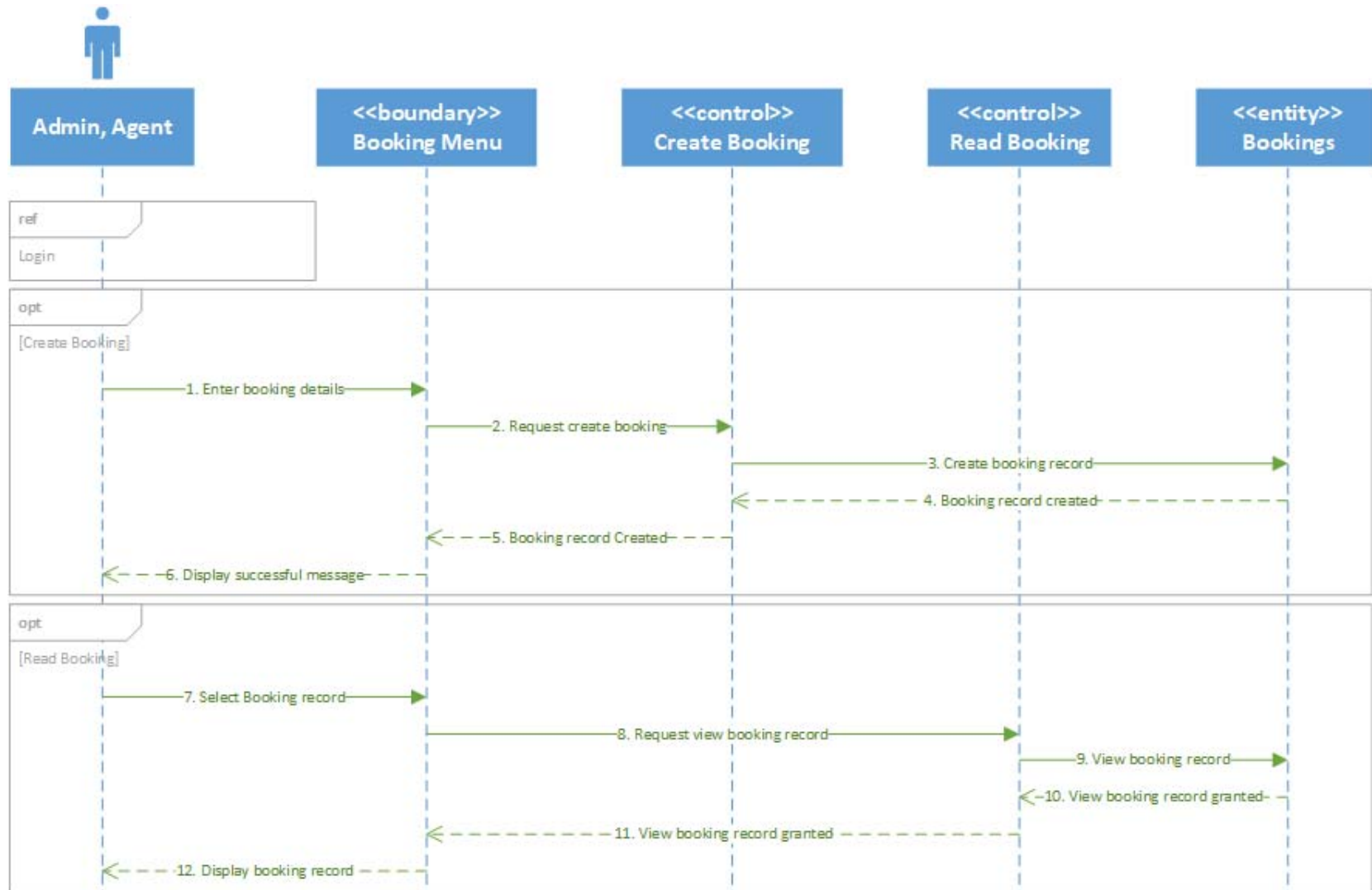
3.3.3.2 Register Agent



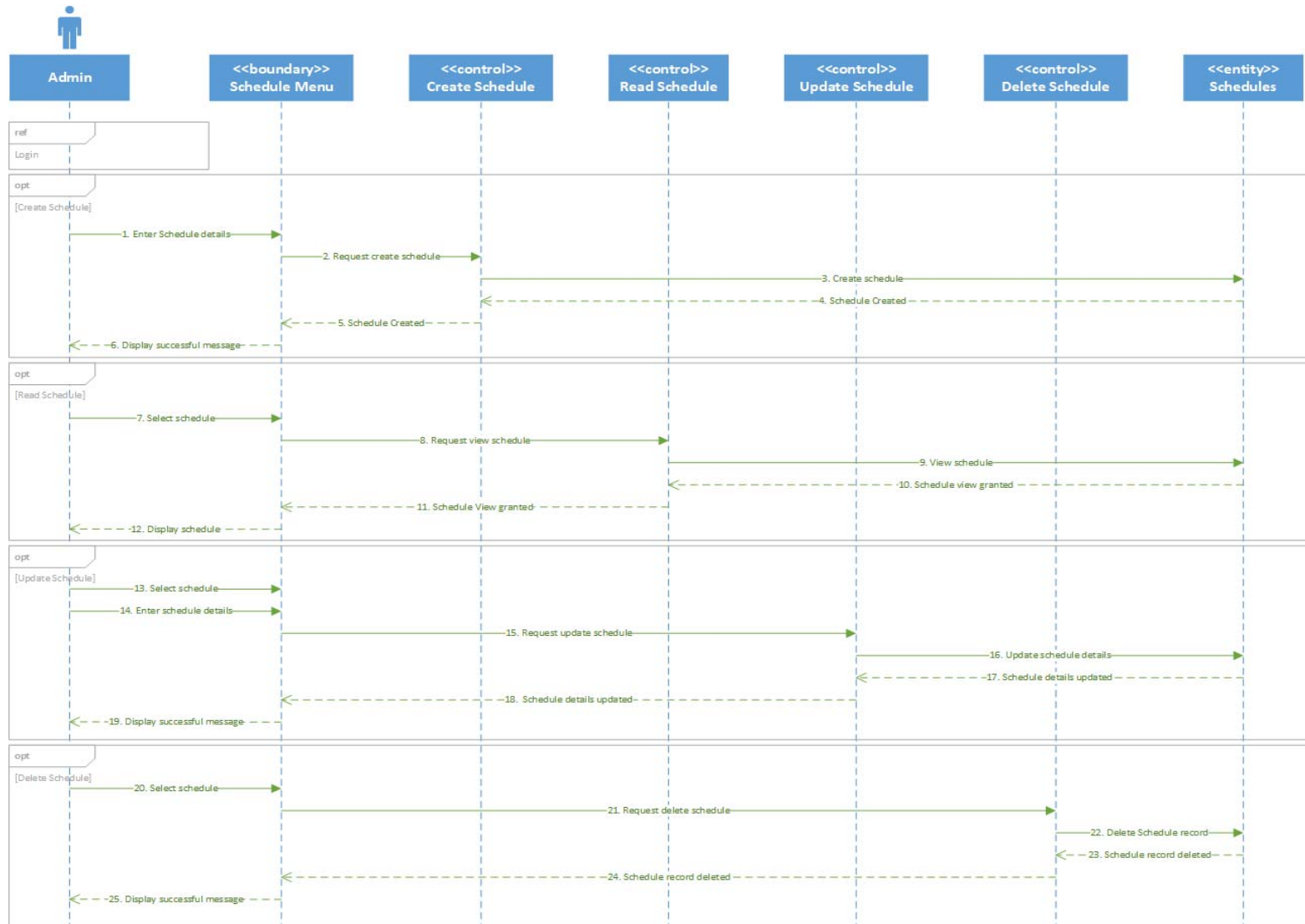
3.3.3.3 Manage Customer



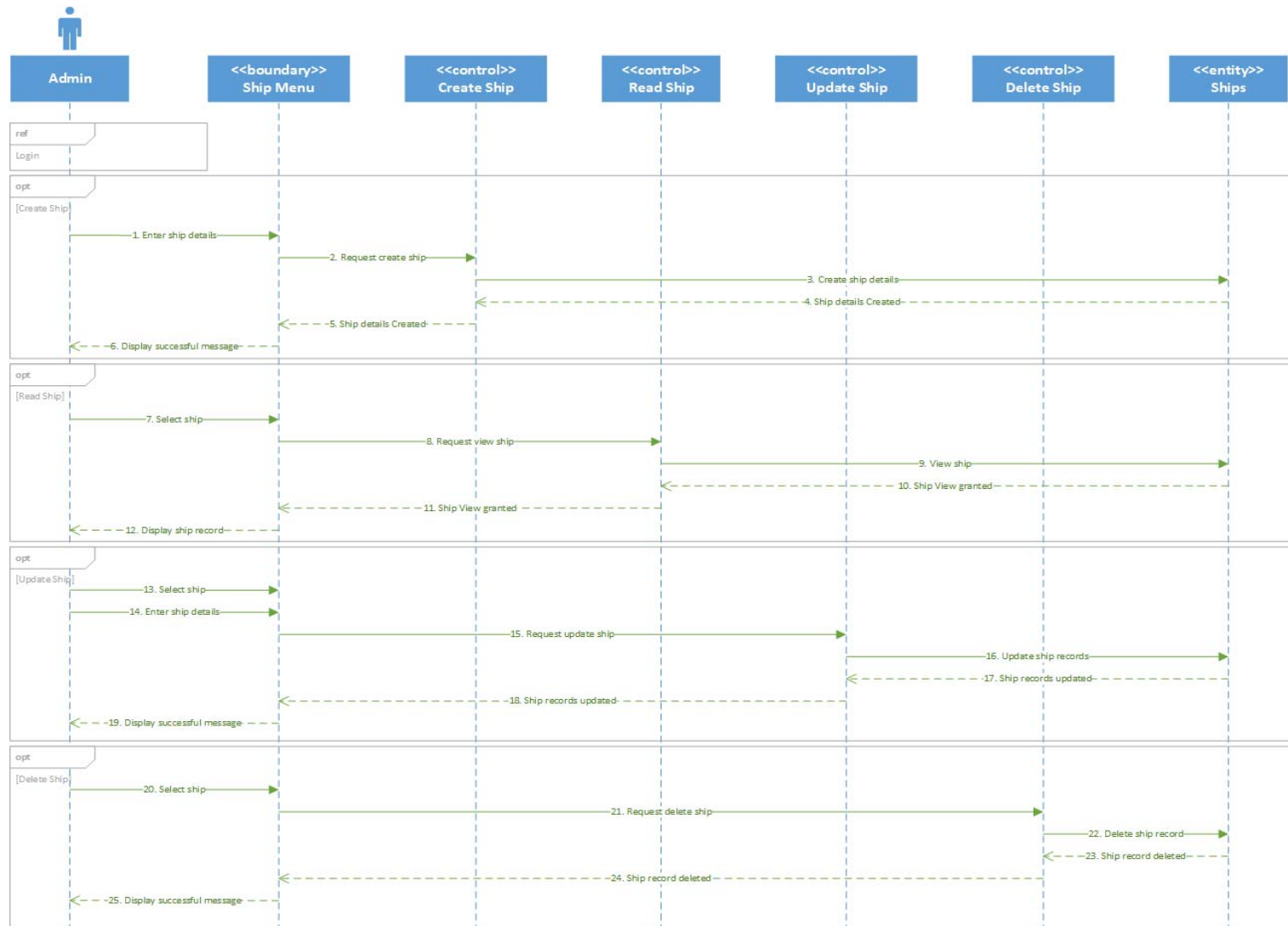
3.3.3.4 Manage Booking



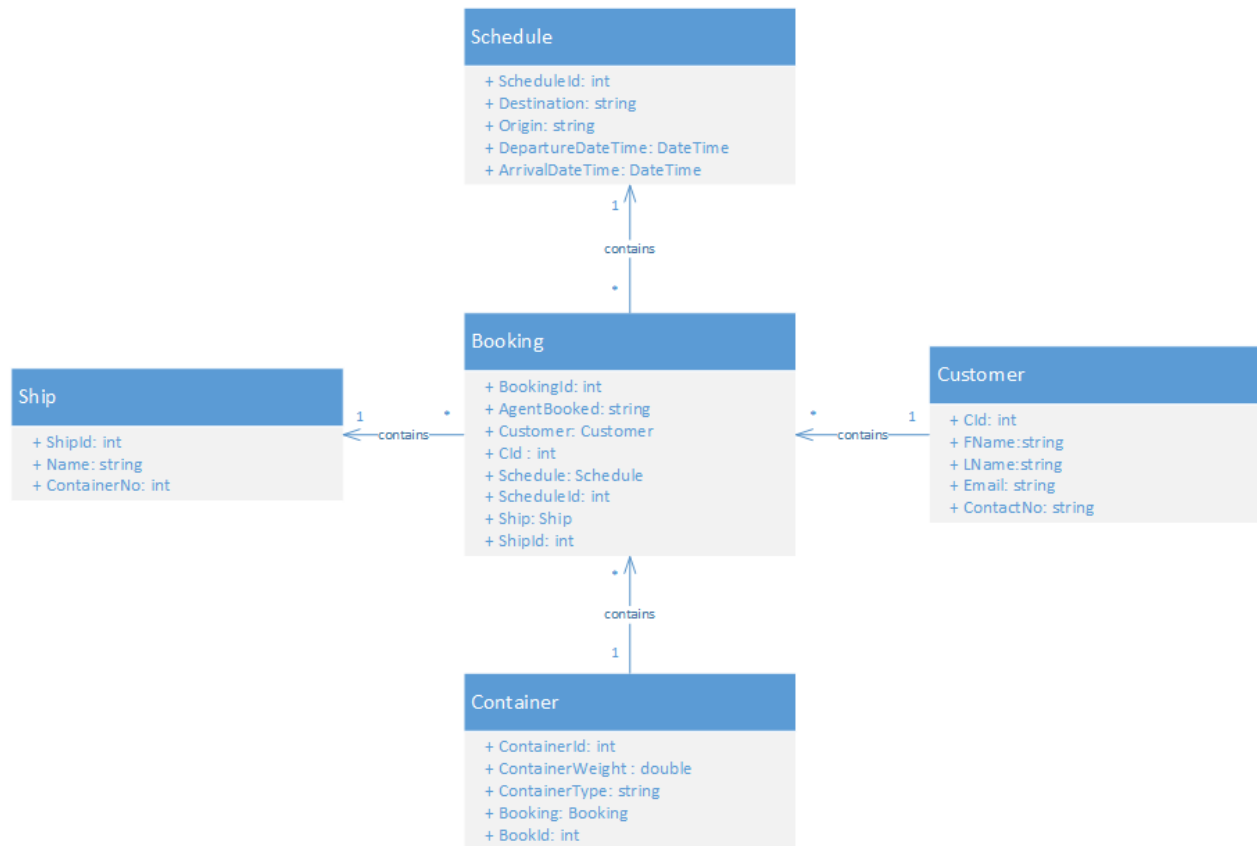
3.3.3.5 Manage Schedule



3.3.3.6 Manage Ship



3.3.4 Class Diagram



CHAPTER 4: IMPLEMENTATION

4.1 ASP.NET Web Application

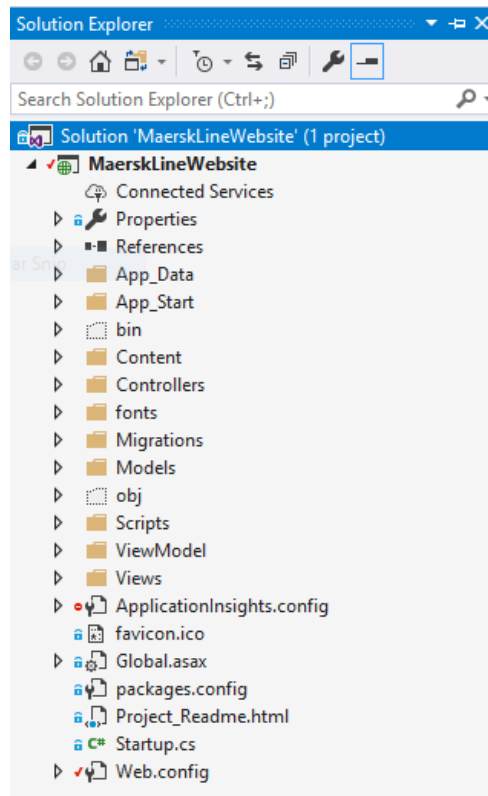


Figure 3 MaerskLineWebsite Project in Visual Studio 2017

MVC framework is known as the **Model-View-Controller** architectural pattern and it provides alternatives to ASP.NET Web Forms pattern to create Web Application. MaerskLine CMS used MVC framework to develop the system. Based on the figure 3, it separates into 3 main categories; model, controller and view. The model is defined as the parts of application that develop the logic for application's data domain and the model object get and store model state in database. View is defined as the components that shows the user interface of the application. Controller works and handles both user interaction and model. It also able to choose views to render that display user interfaces (Microsoft, 2018).

In MaerskLine CMS, it includes Bootstrap 3 as well where most of the application themes with different designs can be found. It also able to adjust the website based on the various screen sizes.

4.2 Azure Resource Group

<input type="checkbox"/> NAME ↑↓	SUBSCRIPTION ↑↓	LOCATION ↑↓	
<input type="checkbox"/> DB-NCUS-MaerskLine	Free Trial	North Central US	...
<input type="checkbox"/> DB-SEA-MaerskLine	Free Trial	Southeast Asia	...
<input type="checkbox"/> NCUS-MaerskLine	Free Trial	North Central US	...
<input type="checkbox"/> SEA-MaerskLine	Free Trial	Southeast Asia	...
<input type="checkbox"/> TrafficManager-MaerskLine	Free Trial	Southeast Asia	...

Figure 4 Azure Resource Group (MaerskLine)

4.2.1 Resource Group of Web Application Service

Resource group
Create an empty resource group

* Resource group name
maerskline-SEA ✓

* Subscription
Free Trial ▼

* Resource group location
Southeast Asia ▼

Create

Resource group
Create an empty resource group

* Resource group name
maerskline-ncus ✓

* Subscription
Free Trial ▼

* Resource group location
North Central US ▼

Create

Figure 5 Creating 2 different regions of web application service (SEA and NCUS)

In Maersk Line, there are 2 resource groups that related to web application service are created for 2 different regions; Southeast Asia (SEA) and North Central US (NCUS). The primary (SEA) and secondary (NCUS) web app will be created with the help of resource group. The resource group name for SEA is “maerskline-SEA” whereas the resource group name for NCUS is “maerskline-NCUS”.

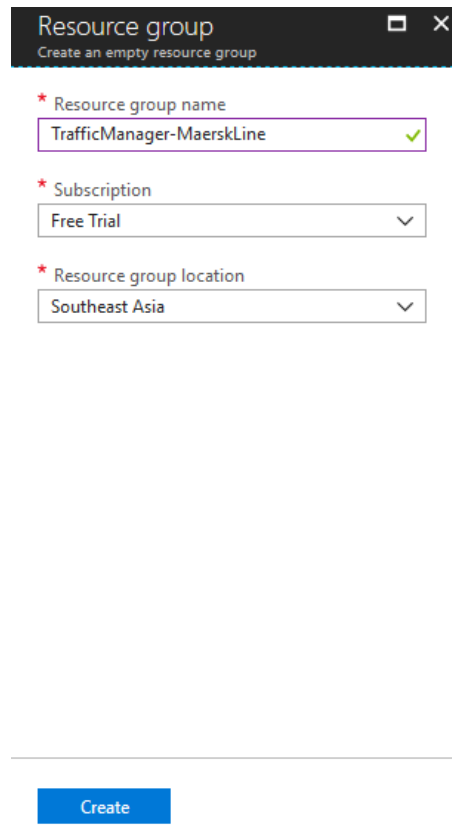
4.2.2 Resource Group of SQL Database

Resource group name	Subscription	Resource group location
DB-NCUS-MaerskLine	Free Trial	North Central US
DB-SEA-MaerskLine	Free Trial	Southeast Asia

Figure 6 Creating 2 different regions of SQL Database (SEA and NCUS)

In Maersk Line, there are 2 resources groups that related to SQL Database are created for 2 different regions; Southeast Asia (SEA) and North Central US (NCUS). The primary (SEA) and secondary (NCUS) SQL database will be created with the help of resource group. The resource group name for SEA is “DB-SEA-maerskline” whereas the resource group name for NCUS is “DB- NCUS-maerskline”. The resource groups for SQL database are created to maintain the database services to meet user’s requirements.

4.2.3 Resource Group of Traffic Manager



Resource group
Create an empty resource group

* Resource group name
TrafficManager-MaerskLine ✓

* Subscription
Free Trial ▼

* Resource group location
Southeast Asia ▼

Create

Figure 7 Creating traffic manager

Besides, a resource group of traffic manager is created as well for later use. The name of the resource group is “TrafficManager-MaerskLine”.

4.3 Azure SQL Server

<input type="checkbox"/>	NAME	TYPE	RESOURCE GROUP	LOCATION	SUBSCRIPTION	
<input type="checkbox"/>	maerskline-ncus-server	SQL server	DB-NCUS-MaerskLine	North Central US	Free Trial	...
<input type="checkbox"/>	maerskline-sea-server	SQL server	DB-SEA-MaerskLine	Southeast Asia	Free Trial	...

Figure 8 Azure SQL Server

4.3.1 Setup SQL Servers

Figure 9 Setup SQL Server for both regions (SEA and NCUS) (pt.1)

The image displays two side-by-side screenshots of the Azure portal's 'SQL Server (logical server o...)' configuration page. The left screenshot shows the configuration for 'maerskline-ncus-server' with fields for server name, admin login (magpie0511), password, confirm password, subscription (Free Trial), resource group (DB-NCUS-MaerskLine), and location (North Central US). The right screenshot shows the same configuration page with the 'Server admin login' field highlighted, showing 'magpie0511' and a green checkmark. Both screenshots have a 'Create' button and an 'Automation options' link at the bottom.

Figure 10 Setup SQL Server for both regions (SEA and NCUS) (pt.2)

SQL server is needed to be implemented before setting up SQL Database. It is needed to host the database into the cloud later. The server name for SEA is “maerskline-sea-server.database.windows.net” whereas for NCUS is “maerskline-ncus-server.database.windows.net”. It requires server admin login and password for security purpose. The resource group of SQL database that created earlier are used to create the SQL Server.

4.4 Azure SQL Database

4.4.1 Setup Primary SQL Database

The image displays two side-by-side screenshots of the Azure SQL Database setup wizard. The left window shows the 'Database name' field with 'maerskline-sea-db', 'Subscription' as 'Free Trial', 'Resource group' as 'DB-SEA-MaerskLine', 'Select source' as 'Blank database', and 'Server' as 'maerskline-sea-server (Southe...'. The right window shows 'Free Trial' for subscription, 'DB-SEA-MaerskLine' for resource group, 'Blank database' for source, 'maerskline-sea-server (Southe...' for server, 'Standard S1: 20 DTUs, 250 GB' for pricing tier, and 'SQL_Latin1_General_CP1_CI_AS' for collation. Both windows have 'Create' and 'Automation options' buttons at the bottom.

Figure 11 Setup Primary SQL Database

In this step, a SEA SQL database will be created as it is the primary region. The SQL Database is created with the database name “maerskline-sea-db”. It involves with the 2 main components; “DB-SEA-MaerskLine” resource group and the SQL server, “maerskline-sea-server”.

Resource Configuration & Pricing

Feedback

Basic
For less demanding workloads
Starting at 20.96 MYR / month

Standard
For most production workloads
Starting at 63.00 MYR / month

Premium
For IO-intensive workloads.
Not available
Starting at 1953.00 MYR / month

vCore-based purchasing options
[Preview] Click here to customize your performance using vCores

DTUs What is a DTU?
10 50 100 200 400 800 1600 3000 20 (\$1)

Max data size
100 MB 250 GB 250 GB

Cost Summary
Cost per DTU (in MYR) 6.30
DTUs selected x 20
EST. COST PER MONTH 125.99 MYR

Apply

Figure 12 Resource Configuration and Pricing

Based on Figure 12, the pricing tier of the Primary SQL Database is “Standard S1: 20 DTUs, 250GB”. DTU is stand for Database Throughput Unit which involved a mixture measure of CPU, memory, reads and writes. The DTU-purchasing model requires users to choose either **Basic**, **Standard** and **Premium** service tiers for the single databases and elastic pools, where these service tiers can be differentiated by different levels of performance range with fixed amount of included storage, fixed retention period for backups and fixed cost (Azure, 2018).

4.4.2 Setup Secondary SQL Database

4.4.2.1 Replicate Secondary Database

Search (Ctrl+/)

Select a region on the map or from the Target Regions list to create a secondary database.

SERVER/DATABASE	FAILOVER POLICY	STATUS
PRIMARY		
Southeast Asia maerskline-sea-server/maerskline-sea-db	None	Online
SECONDARIES		
Geo-Replication is not configured		

Figure 13 Replicate Secondary Database using Geo-replication

A standard geo-replication able to provides users recovery solution targeting application with moderate update rates and interested in reducing downtime of the system (Petrossian, 2014). It is a service to implement a similar database for different region and used for secondary database to replace as the primary database when it is not accessible.

The image displays two side-by-side screenshots of the 'Create secondary' form in Azure. The left screenshot shows the form with 'North Central US' selected for the region, 'maerskline-sea-db' for the database name, 'Readable' for the secondary type, 'maerskline-ncus-server (North...)' for the target server, 'None' for the elastic database pool, and 'Standard S1: 20 DTUs, 250 GB' for the pricing tier. The right screenshot shows the same form with the same selections, but the 'Target server' field is highlighted with a dashed blue border. Both forms have a 'Pin to dashboard' checkbox and an 'OK' button at the bottom.

Figure 14 Create Secondary Database

A create secondary form is displayed after selecting the region on the map respectively. The region “North Central US” is selected and the targeted server is “maerskline-ncus-server”. There will be no elastic database pool and the pricing tier will be “Standard S1: 20 DTUs, 250GB” as well. In Figure 15, the replication of secondary database is in seeding process. Hence, dotted line appeared on the map. In Figure 16, the seeding process is completed so a solid line displayed on the map.



Figure 15 Geo replication (before)



Figure 16 Geo replication (after)

4.4.2.2 Configure Failover Policy

Failover group

Create a failover group to automatically failover databases in it.

Primary server
maerskline-sea-server (southe...

* Secondary server
maerskline-ncus-server (North...

* Failover group name
maerskline-failover
...database.windows.net

Read/Write failover policy
Automatic

Read/Write grace period (hours)
1 hours

Summary
Number of new standalone databases 1
Number of new elastic pools 0

Create

Figure 17 Creating a Failover Group

A failover group is created for the geo-replication by going to the primary database of SQL Server. Then, click on the “Add Group” which is under the “Failover Group”. The failover group name is “maerskline-failover” and the read or write failover policy is automatic. In figure 18, the failover policy is set to the primary database.

PRIMARY			
	Southeast Asia	maerskline-sea-server/maerskline-sea-db	maerskline-failover (Automatic, ... Online
SECONDARIES			
	North Central US	maerskline-ncus-server/maerskline-sea-db	Readable ...

Figure 18 Geo-replication of primary database

4.5 Azure Web Application Service

<input type="checkbox"/>	NAME ↑↓	TYPE ↑↓	RESOURCE GROUP ↑↓	LOCATION ↑↓	SUBSCRIPTION ↑↓	
<input type="checkbox"/>	tp035869-maerskline-ncus	App Service	NCUS-MaerskLine	North Central US	Free Trial	...
<input type="checkbox"/>	tp035869-maerskline-sea	App Service	SEA-MaerskLine	Southeast Asia	Free Trial	...

Figure 19 Azure Web Application Service

4.5.1 Setup Web Application Service

The screenshot shows two side-by-side panels from the Azure portal. The left panel, titled 'Web App + SQL', is in the 'Create' state. It contains the following fields: 'App name' (tp035869-maerskline-sea), 'Subscription' (Free Trial), 'Resource Group' (SEA-MaerskLine), 'App Service plan/Location' (ServicePlana91d6b0c-9afa(Ce...)), 'SQL Database' (Configure required settings), and 'Application Insights' (On). The right panel, titled 'App Service plan', shows a 'Create new' button and a list of plans. A hand cursor is pointing at the 'ServicePlana91d6b0c-9afa(S1) (New)' plan, which is located in 'Central US'.

Figure 20 Create new Web App Service

The App service that chosen in Maersk Line is “Web App + SQL” as the system involved with SQL server and database. The name of the app is “tp035869-maerskline-sea.azurewebsites.net”, the resource group is “SEA-MaerskLine”, and the SQL database is “maerskline-sea-db”. The SQL Admin Username and Password is required. Based on figure 20, the new service plan is created and named as “SEAServicePlan”. The Pricing Tier of the service plan is S1 (Standard), where is provides a single core, 1.75 GB RAM, 50 GB storage, backup data daily, 5 slots of web app staging and traffic manager. Similar procedure is applied to the secondary web app (NCUS) with different app name “tp035869-maerskline-ncus.azurewebsites.net”.

New App Service Plan
Create a plan for the web app

* App Service plan
SEAServicePlan ✓

* Location
Southeast Asia ▼

* Pricing tier
S1 Standard >

Figure 21 New App Service Plan

Choose your pricing tier
Browse the available plans and their features

S1 Standard		S2 Standard		S3 Standard	
1 Core		2 Core		4 Core	
1.75 GB RAM		3.5 GB RAM		7 GB RAM	
50 GB Storage		50 GB Storage		50 GB Storage	
Custom domains / SSL SNI Incl & IP SSL Support		Custom domains / SSL SNI Incl & IP SSL Support		Custom domains / SSL SNI Incl & IP SSL Support	
Up to 10 instance(s) Auto scale		Up to 10 instance(s) Auto scale		Up to 10 instance(s) Auto scale	
Daily Backup		Daily Backup		Daily Backup	
5 slots Web app staging		5 slots Web app staging		5 slots Web app staging	
Traffic Manager Geo availability		Traffic Manager Geo availability		Traffic Manager Geo availability	
937.44 MYR/MONTH (ESTIMATED)		1,874.88 MYR/MONTH (ESTIMATED)		3,749.76 MYR/MONTH (ESTIMATED)	
312.48 MYR/MONTH (ESTIMATED)		624.96 MYR/MONTH (ESTIMATED)		1,249.92 MYR/MONTH (ESTIMATED)	

B1 Basic		B2 Basic		B3 Basic	
1 Core		2 Core		4 Core	
1.75 GB RAM		3.5 GB RAM		7 GB RAM	
10 GB Storage		10 GB Storage		10 GB Storage	
Custom domains		Custom domains		Custom domains	
SSL Support SNI SSL Included		SSL Support SNI SSL Included		SSL Support SNI SSL Included	
Up to 3 instance(s) Manual scale		Up to 3 instance(s) Manual scale		Up to 3 instance(s) Manual scale	
234.36 MYR/MONTH (ESTIMATED)		468.72 MYR/MONTH (ESTIMATED)		937.44 MYR/MONTH (ESTIMATED)	

F1 Free		D1 Shared*	
- Shared infrastructure		- Shared infrastructure	

Select

Figure 22 Pricing Tier.

4.5.2 Deploy APS.NET Web Application

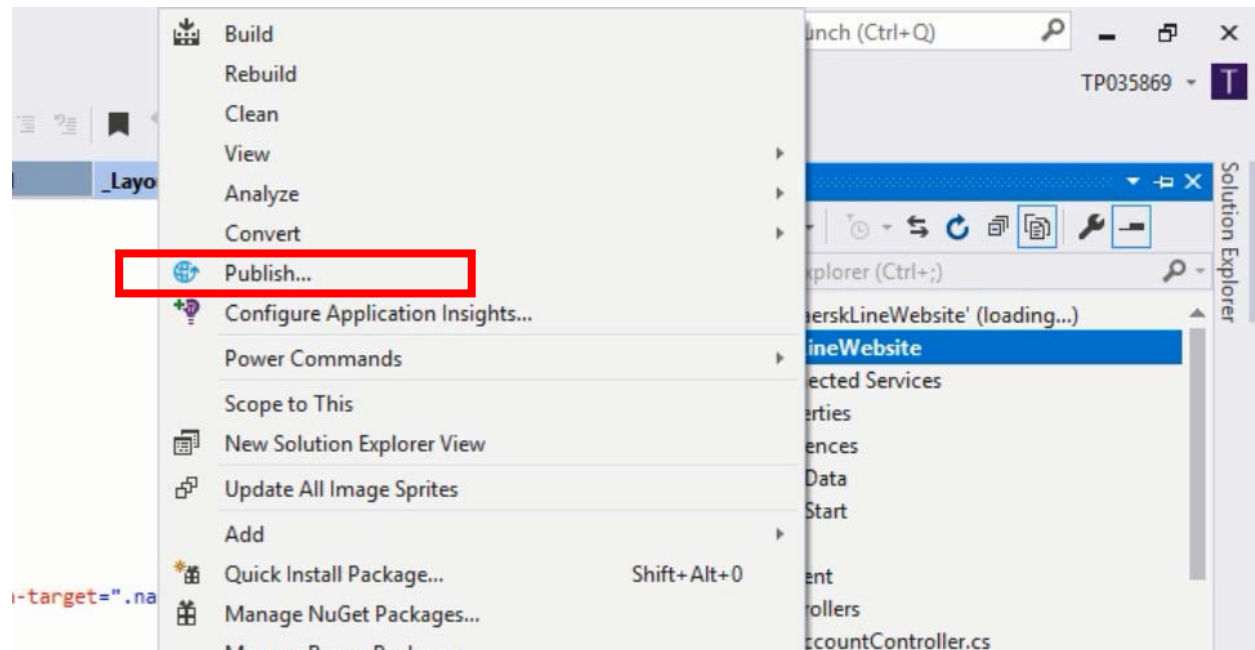


Figure 23 Publishing of Web Application

Once the all the setup of Azure is completed, the Web application can be published to Azure Cloud Platform by right-clicking of the project name and click on “Publish...”.

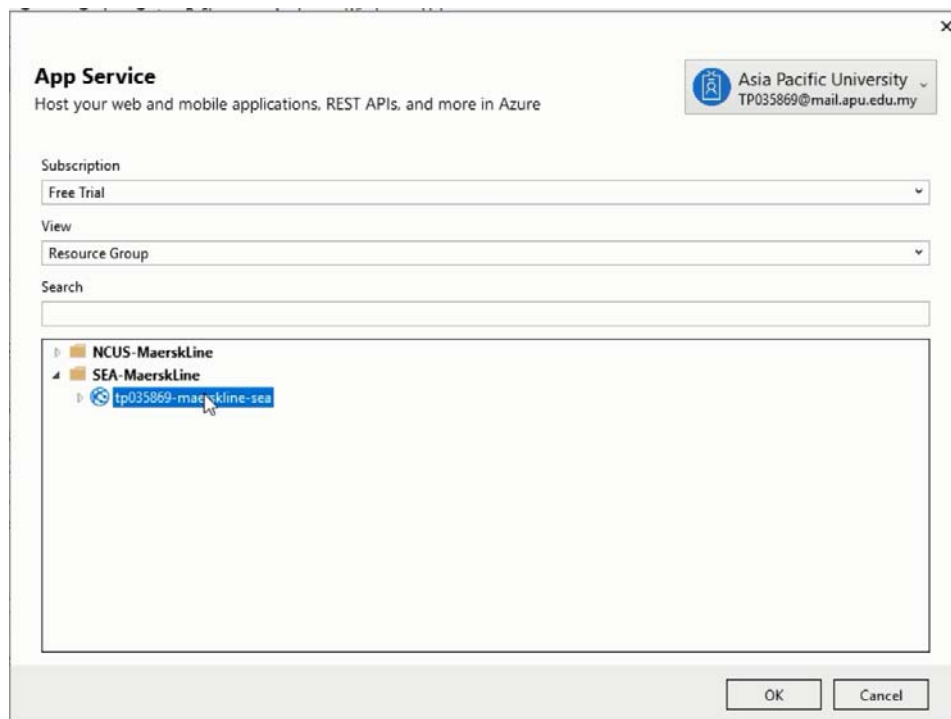


Figure 24 Choose App Service

The App Service is chosen and selected in the next step. Then, click on “OK” button.

The screenshot shows a 'Publish' dialog box with a title bar containing a question mark and a close button. The main area has a 'Publish' header with a globe icon. Below this, there are two tabs: 'Connection' (selected) and 'Settings'. The 'Connection' tab displays the title 'tp035869-maerskline-sea - Web Deploy *'. The 'Publish method' is set to 'Web Deploy' in a dropdown menu. Below this, there are four text input fields: 'Server' (tp035869-maerskline-sea.scm.azurewebsites.net:443), 'Site name' (tp035869-maerskline-sea), 'User name' (\$tp035869-maerskline-sea), and 'Password' (masked with dots). A 'Save password' checkbox is checked. Below the password field is a 'Destination URL' field containing 'http://tp035869-maerskline-sea.azurewebsites.net'. A 'Validate Connection' button is present, with a green checkmark icon to its right. At the bottom of the dialog, there are four buttons: '< Prev', 'Next >', 'Save', and 'Cancel'. A mouse cursor is pointing at the 'Next >' button.

Figure 25 Connection Settings

The settings will be checked. Once completed, the click on “Validate Connection”. When there is a green tick beside the button, it means the connection is ready and is able to proceed to the next step.

The screenshot shows the 'Publish' dialog box in Visual Studio. The 'Connection' tab is selected, and the 'Settings' sub-tab is active. The connection name is 'tp035869-maerskline-sea - Web Deploy *'. The 'Configuration' is set to 'Release'. Under 'File Publish Options', the 'Databases' section is expanded, showing 'ApplicationDbContext (DefaultConnection)'. The 'Data Source' is 'tcp:maerskline-sea-server.database.windows.net,1433;Initial Cat'. There are two checkboxes: 'Use this connection string at runtime (update destination web.config)' and 'Execute Code First Migrations (runs on application start)'. Both are checked. At the bottom, there are buttons for '< Prev', 'Next >', 'Save', and 'Cancel'.

Figure 26 Database Settings

The “Execute Code First Migrations (runs on application start)” is selected because the project used code-first migrations to store data into the database. Then, the click on the 3 dots that is beside the connection string. In Figure 27, the user name and password is required to enter into the form and click on the “test connection”. Once test connection succeeded, the web application is deployed successfully. The figure 28 and 29 show the results and the URL for both SEA and NCUS region.

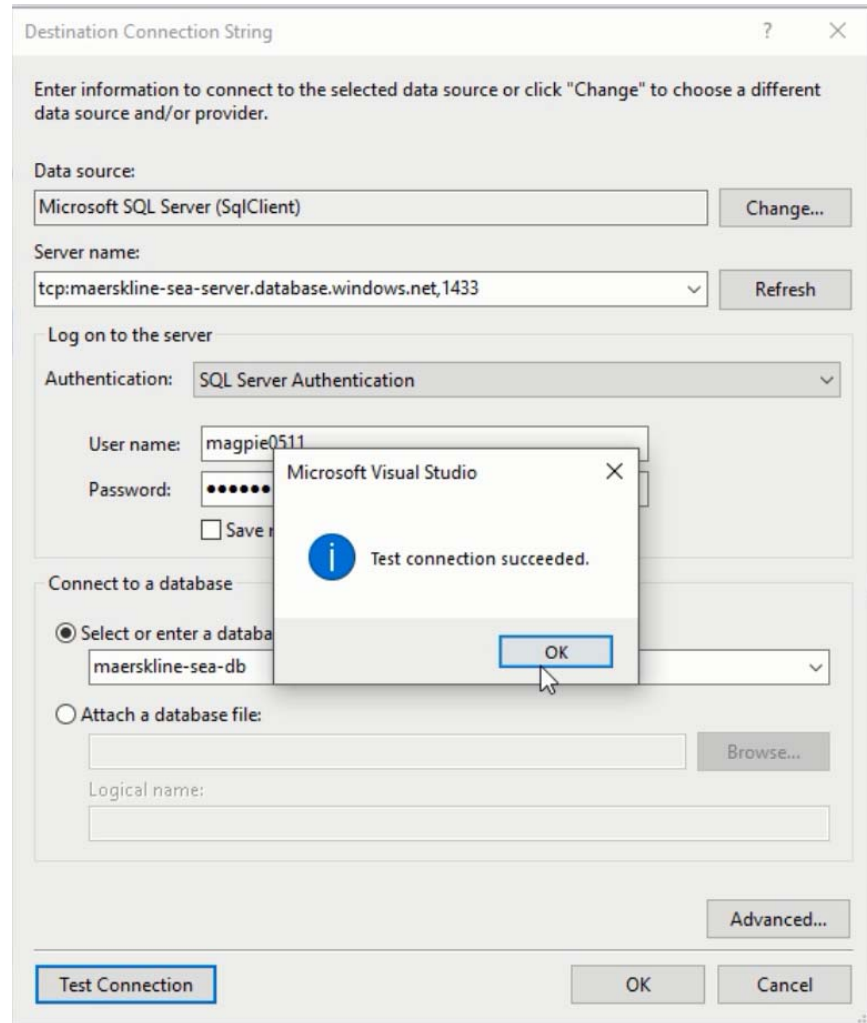


Figure 27 Test Connection Succeeded.

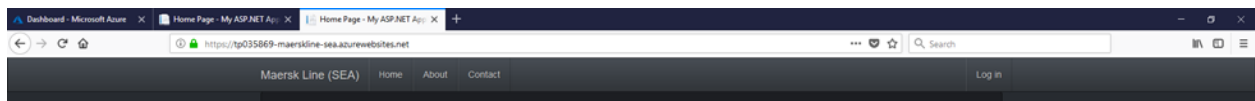


Figure 28 Web application in SEA region

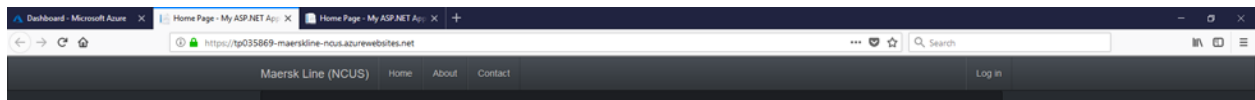
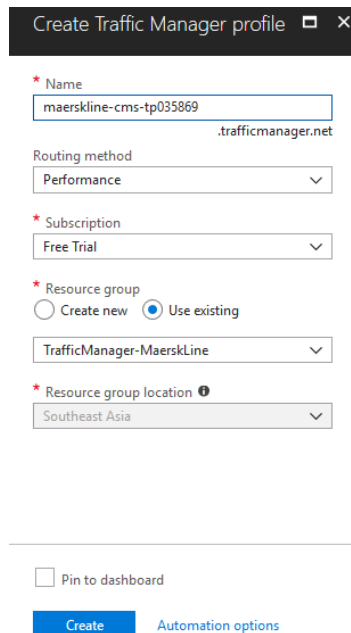


Figure 29 Web application in NCUS region

4.6 Azure Traffic Manager

4.6.1 Setup Traffic Manager Profile



The screenshot shows the 'Create Traffic Manager profile' form in the Azure portal. The form is titled 'Create Traffic Manager profile' with a close button. It contains the following fields and options:

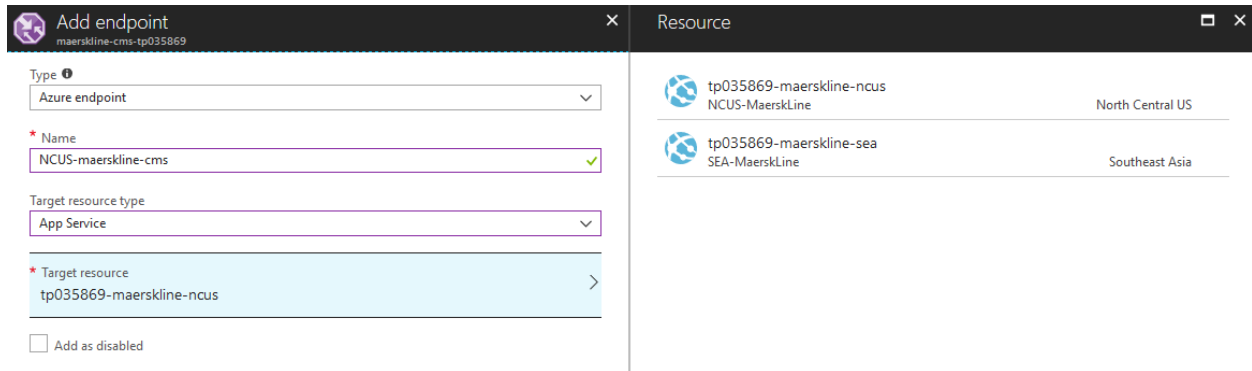
- Name:** A text input field containing 'maerskline-cms-tp035869'. Below the field, the text '.trafficmanager.net' is displayed.
- Routing method:** A dropdown menu with 'Performance' selected.
- Subscription:** A dropdown menu with 'Free Trial' selected.
- Resource group:** Radio buttons for 'Create new' and 'Use existing'. 'Use existing' is selected. Below the radio buttons, a dropdown menu shows 'TrafficManager-MaerskLine'.
- Resource group location:** A dropdown menu with 'Southeast Asia' selected.

Below the form, there is a checkbox for 'Pin to dashboard' which is unchecked. At the bottom, there is a blue 'Create' button and a link for 'Automation options'.

Figure 30 Create Traffic Manager Profile

Search for “Traffic Manager Profile” in create a resource. Then, the name of the traffic manager profile will be “maerskline-cms-tp035869.trafficmanager.net” and the resource group is from “TrafficManager-MaerskLine”. The routing method is based on the performance of the network latency.

4.6.2 Setup Endpoints



Add endpoint
maerskline-cms-tp035869

Type: Azure endpoint

* Name: NCUS-maerskline-cms

Target resource type: App Service

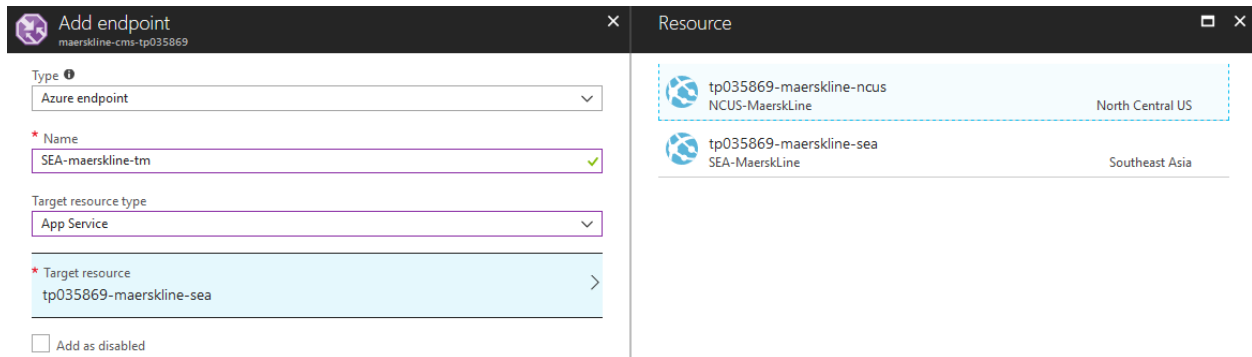
* Target resource: tp035869-maerskline-ncus

☐ Add as disabled

Resource

tp035869-maerskline-ncus NCUS-MaerskLine	North Central US
tp035869-maerskline-sea SEA-MaerskLine	Southeast Asia

Figure 31 Creating endpoint of NCUS



Add endpoint
maerskline-cms-tp035869

Type: Azure endpoint

* Name: SEA-maerskline-tm

Target resource type: App Service

* Target resource: tp035869-maerskline-sea

☐ Add as disabled

Resource

tp035869-maerskline-ncus NCUS-MaerskLine	North Central US
tp035869-maerskline-sea SEA-MaerskLine	Southeast Asia

Figure 32 Creating endpoint of SEA

The URL of the Web Apps will be the endpoints of the traffic manager. In the target resource, the web app URL will be selected based on the given region. The target resource type will be “App Service” and the name of the endpoints are “SEA-maerskline-tm” and “NCUS-maerskline-cms” respectively.

4.6.3 Testing Traffic Manager Profile

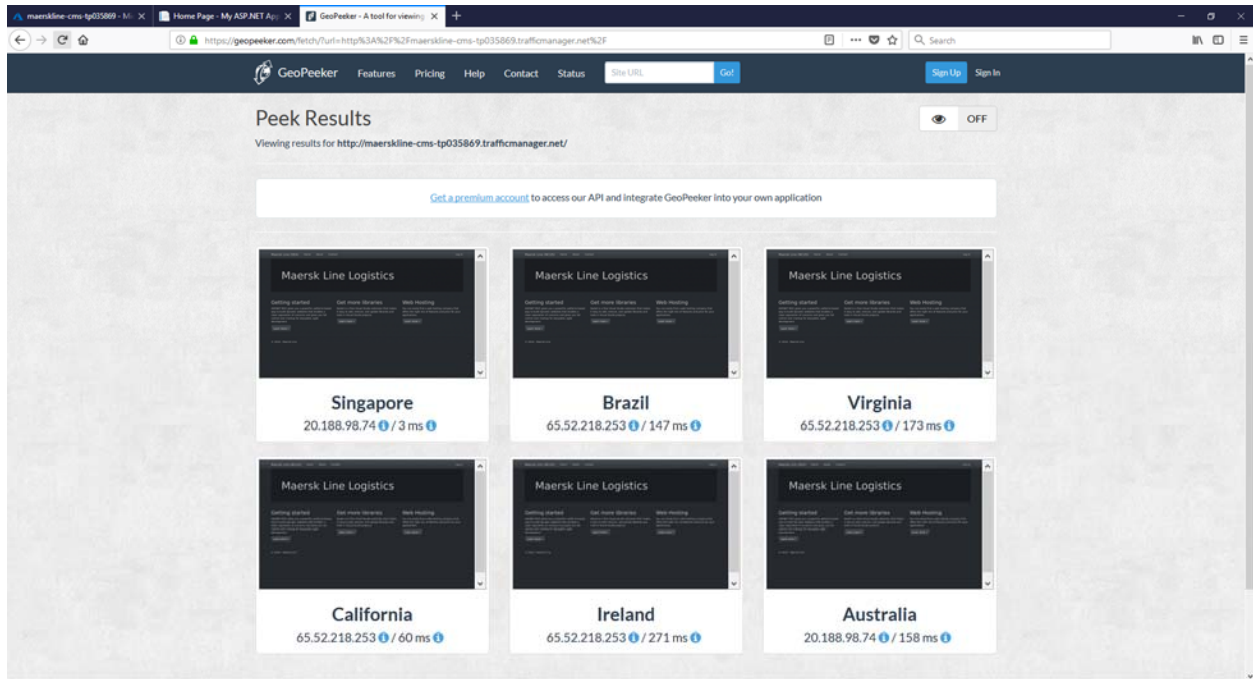


Figure 33 Testing of Traffic Manager Profile

GeoPeeker will be used to test the URL of Traffic Manager Profile by using IP address to access from different countries; Singapore, Brazil, Virginia, California, Ireland and Australia.

4.7 Azure Web Application Autoscaling

4.7.1 Setup Web Application Scale Out

[Configure](#) [Run history](#) [JSON](#) [Notify](#)

* Autoscale setting name

Resource group

Default Auto created scale condition

Delete warning

The very last or default recurrence rule cannot be deleted. Instead, you can disable autoscale to turn off autoscale.

Scale mode ☒ Scale based on a metric ☐ Scale to a specific instance count

It is recommended to have at least one scale in rule

Scale out

When	SEAServicePlan	(Average) MemoryPercentage > 75	Increase instance count by 1
Or	SEAServicePlan	(Average) CpuPercentage > 70	Increase instance count by 1

+ Add a rule

Instance limits

Minimum Maximum Default

Schedule

This scale condition is executed when none of the other scale condition(s) match

+ Add a scale condition

Figure 34 Create Web Application Scale Out

4.7.2 Setup Scale Rule for Auto Scaling Plan

The figure displays two identical 'Scale rule' configuration windows side-by-side. Each window has a title bar with 'Scale rule' and a close button. The configuration is organized into several sections:

- Metric source:** A dropdown menu set to 'Current resource (SEAServicePlan)'.
- Resource type:** A dropdown menu set to 'App Service plans'.
- Resource:** A dropdown menu set to 'SEAServicePlan'.
- Criteria:**
 - * Time aggregation:** A dropdown menu set to 'Average'.
 - * Metric name:** A dropdown menu. In the left window, it is 'CPU Percentage'; in the right window, it is 'Memory Percentage'. A '1 minute time grain' label is visible to the right of the dropdown.
 - * Time grain statistic:** A dropdown menu set to 'Average'.
 - * Operator:** A dropdown menu set to 'Greater than'.
 - * Threshold:** A text input field. In the left window, it is '70'; in the right window, it is '75' with a green checkmark and a '%' symbol to its right.
 - * Duration (in minutes):** A text input field set to '10' in the left window and '5' in the right window, both with green checkmarks.
- Action:**
 - * Operation:** A dropdown menu set to 'Increase count by'.
 - * Instance count:** A text input field set to '1'.
 - * Cool down (minutes):** A text input field set to '5'.

At the bottom of each window is a blue 'Add' button.

Figure 35 Setup Scale Rule for Auto Scaling Plan

The scale rule is used to setup for auto scaling plan. In this web application, the metric source is named as “Current resource (SEAServicePlan)”. However, the threshold for both Scale Rules are 70 and 75 respectively. Threshold is the memory percentages that run in the web application. When the scale rule with Threshold 70 met, then the web app will increase the scale by one instance, which is up to a maximum number of 3.

4.8 System Interface

4.8.1 Login

The screenshot shows a web browser window with the title 'Log in - My ASP.NET Application'. The address bar displays 'maerskline-cms-tp035869.trafficmanager.net/Account/'. The page header includes 'Maersk Line (SEA)' and navigation links for 'Home', 'About', and 'Contact'. The main content area is titled 'Log in.' and instructs users to 'Use a local account to log in.' It features input fields for 'Email' and 'Password', a 'Remember me?' checkbox, and a 'Log in' button. A link to 'Register as a new user' is located below the login button. The footer contains the copyright notice '© 2018 - Maersk Line'.

Log in - My ASP.NET Application X

maerskline-cms-tp035869.trafficmanager.net/Account/

Maersk Line (SEA) Home About Contact

Log in.

Use a local account to log in.

Email

Password

☐ Remember me?

Log in

[Register as a new user](#)

© 2018 - Maersk Line

Figure 36 System Interface (Login)

4.8.2 Main Page

4.8.2.1 Home Page (Admin)

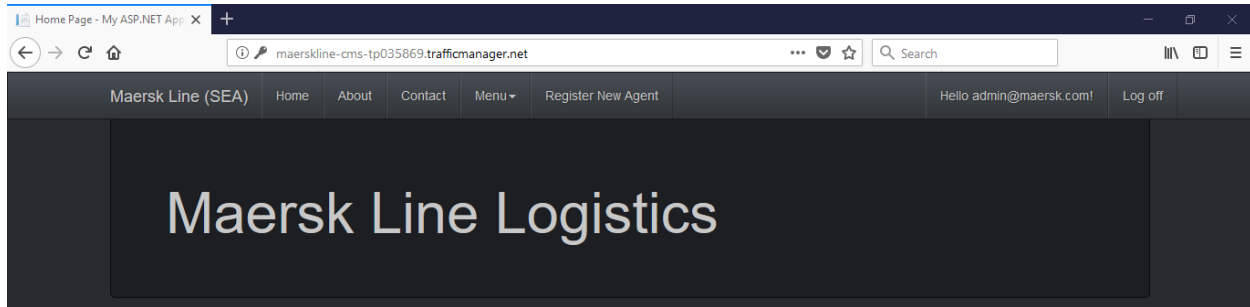


Figure 37 System Interface (Admin Home Page)

4.8.2.2 Home Page (Agent)

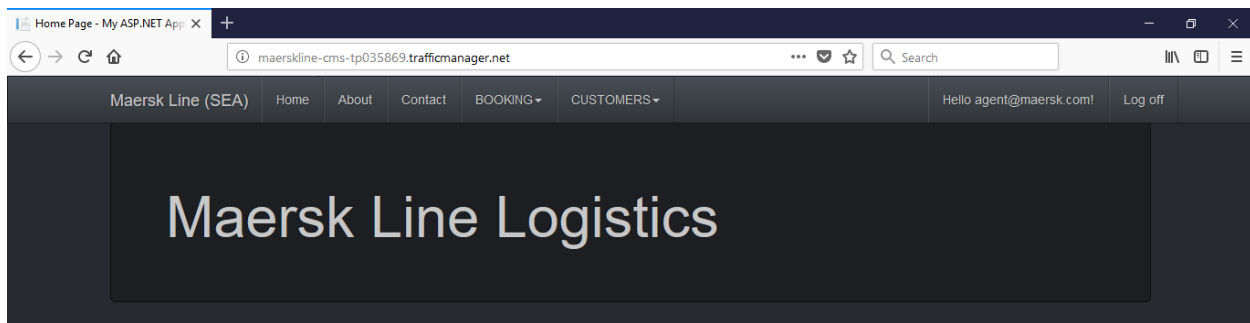


Figure 38 System Interface (Agent Home Page)

4.8.3 Register Agent

Register - My ASP.NET Application X

maerskline-cms-tp035869.trafficmanager.net/Account/Register

Maersk Line (SEA) Home About Contact Menu Register New Agent

Register.

Create a new account.

Email

Password

Confirm password

Register

© 2018 - Maersk Line

Figure 39 System Interface (Register Agent)

4.8.4 Manage Customer

4.8.4.1 Read Customer

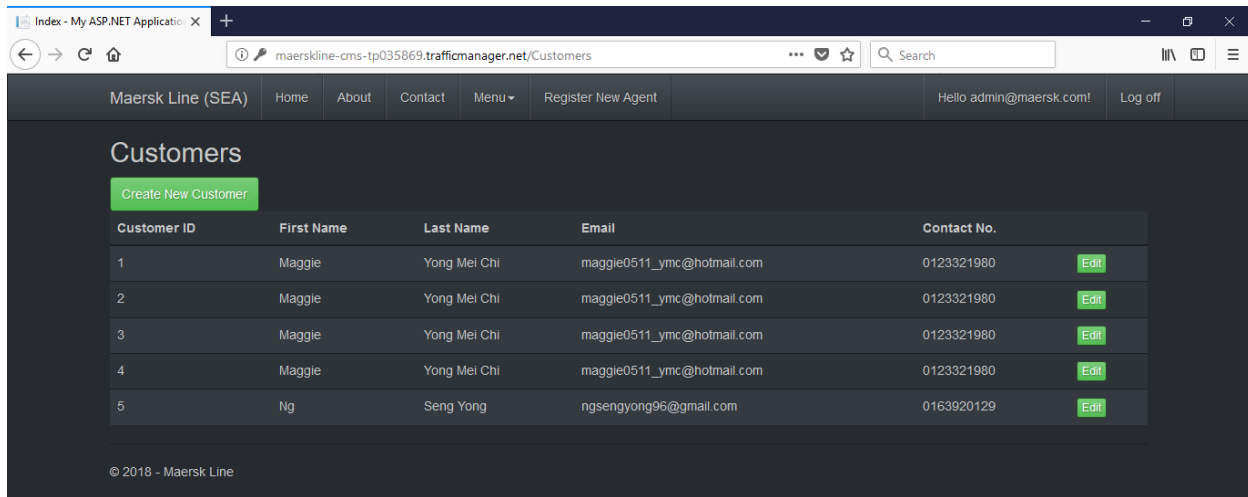


Figure 40 System Interface (Read Customer)

4.8.4.2 Create Customer

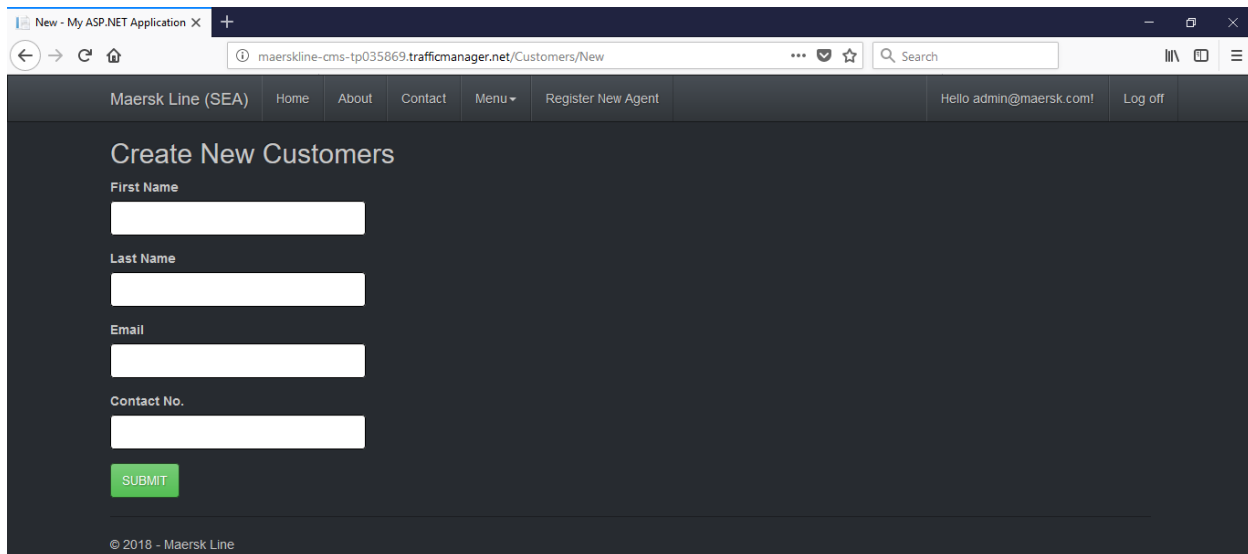


Figure 41 System Interface (Create Customer)

4.8.4.3 Update Customer

Maersk Line (SEA) Home About Contact Menu Register New Agent Hello admin@maersk.com! Log off

Edit Customer

First Name
Maggie

Last Name
Yong Mei Chi

Email
maggie0511_ymc@hotmail.com

Contact No.
0123321980

SAVE

© 2018 - Maersk Line

Figure 42 System Interface (Update Customer)

4.8.5 Manage Booking

4.8.5.1 Read Booking

Booking ID	Booking Agent	Customer ID	Customer First Name	Customer Last Name	Customer Email	Customer Contact No	Ship ID	Ship Name	No. of Container	Container ID	Container Type	Container Weight	Schedule ID	Schedule Origin	Schedule Destination	Schedule Departure Date & Time	Schedule Arrival Date & Time
1	admin@maersk.com	2	Maggie	Yong Mei Chi	maggie0511_ymc@hotmail.com	0123321960	1	Ken	0	1	Solid	100	1	LDN	KUL	4/12/2018 3:37:00 PM	4/28/2018 3:37:00 PM

Figure 43 System Interface (Create Booking)

4.8.5.2 Create Booking

Schedule ID	Schedule Origin	Schedule Destination	Schedule Departure Time	Schedule Arrival Time
1	LDN	KUL	4/12/2018 3:37:00 PM	4/28/2018 3:37:00 PM

Ship ID	Ship Name	No. of Container
1	Ken	0

First Name	Last Name	Email	Contact No
Maggie	Yong Mei Chi	maggie0511_ymc@hotmail.com	0123321960

New Container

Type

Weight (lbs)

Create Booking

Figure 44 System Interface (Update Booking)

4.8.6 Manage Schedule

4.8.6.1 Read Schedule

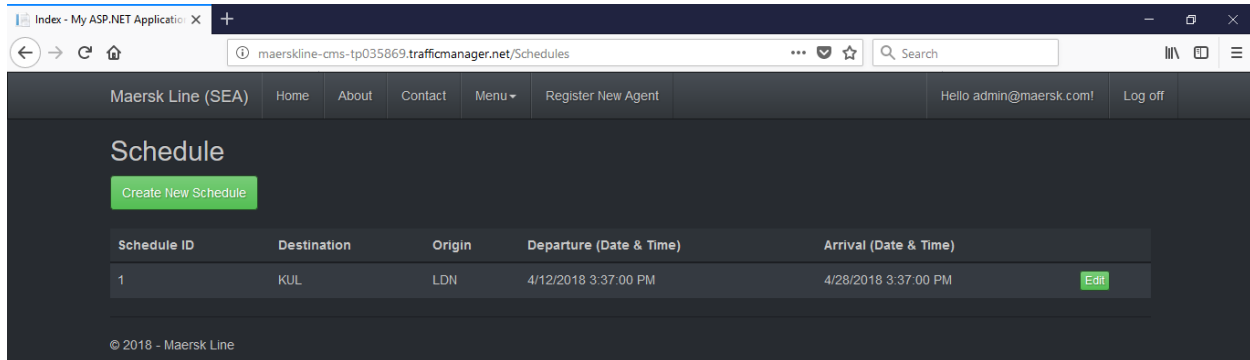


Figure 45 System Interface (Read Schedule)

4.8.6.2 Create Schedule

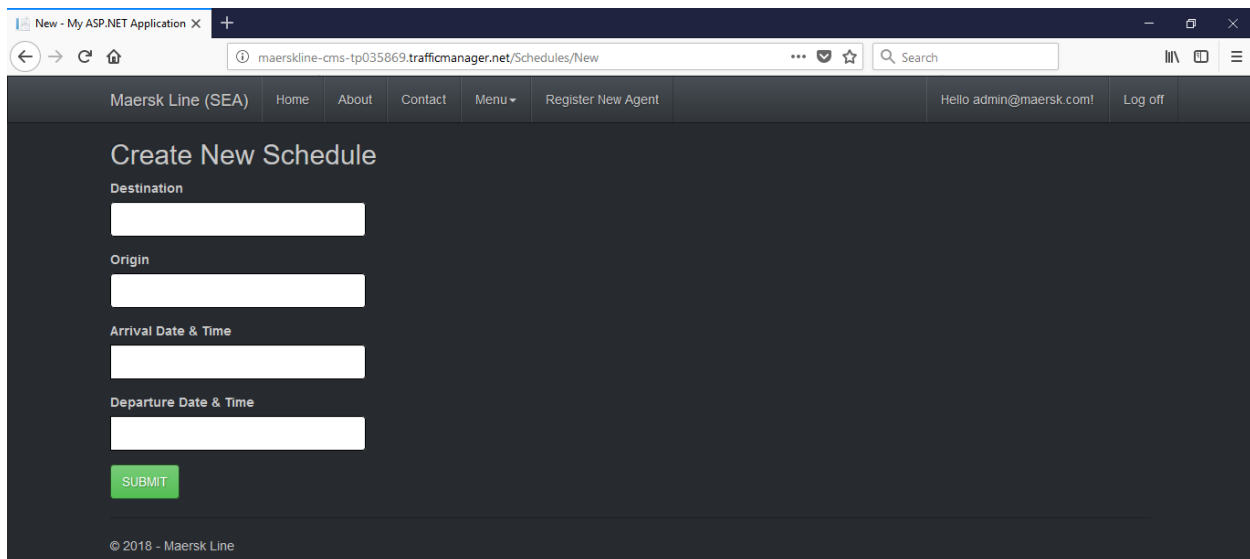


Figure 46 System Interface (Create Schedule)

4.8.6.3 Update Schedule

New - My ASP.NET Application X

maerskline-cms-tp035869.trafficmanager.net/Schedules/Edit/1

Maersk Line (SEA) Home About Contact Menu Register New Agent Hello admin@maersk.com! Log off

Edit Schedule

Destination
KUL

Origin
LDN

Arrival Date & Time
04/28/2018 3:37 PM

Departure Date & Time

SAVE

© 2018 - Maersk Line

Figure 47 System Interface (Update Schedule)

4.8.7 Manage Ship

4.8.7.1 Read Ship

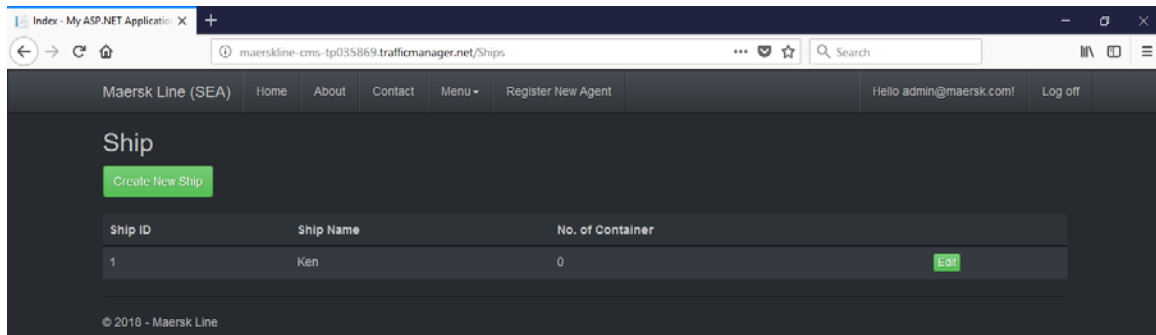


Figure 48 System Interface (Read Ship)

4.8.7.2 Create Ship

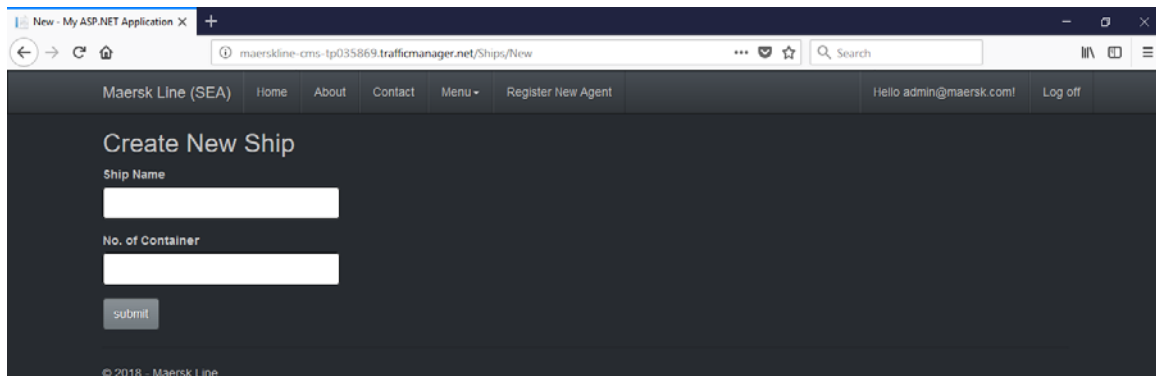


Figure 49 System Interface (Create Ship)

4.8.7.3 Update Ship

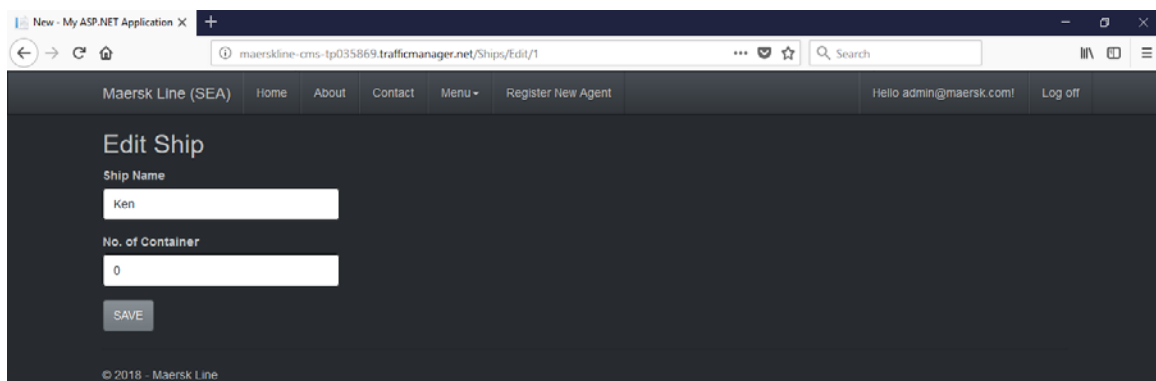


Figure 50 System Interface (Update Ship)

Chapter 5: TESTING

5.1 Performance Testing

Azure has a performance testing function that allows users to test the performance of the application and stimulate local conditions of different regions. The performance testing features is found in the Web Service that created earlier. In this form, the “new” button is clicked. Then, the “ManualTest 1 Url” is selected as test type. The name of the performance test is “PerfTest01”. The generate load from Southeast Asia, which is the Web app Location. The user load for this performance test is 250 and the duration is 5 minutes. In Figure 52, the performance test is completed and it takes about 10 minutes to complete. The status changed to “Completed” when the testing is completed.

The image shows two side-by-side dialog boxes from the Azure portal. The left dialog, titled 'New performance test', has a 'PREVIEW' tab and contains the following fields: 'CONFIGURE TEST USING' with a right arrow, 'Test type: ManualTest 1 Url', 'NAME' with the value 'PerfTest01', 'GENERATE LOAD FROM' with a dropdown menu showing 'Southeast Asia (Web app Location)', 'USER LOAD' with the value '250', and 'DURATION (MINUTES)' with the value '5'. At the bottom is a blue 'Run test' button. The right dialog, titled 'Configure test using', also has a 'PREVIEW' tab and contains: 'TEST TYPE' with a dropdown menu showing 'Manual Test', and 'URL' with the value 'http://maerskline-cms-tp035869.trafficmana...'. At the bottom is a blue 'Done' button.

Figure 51 Create Performance Testing

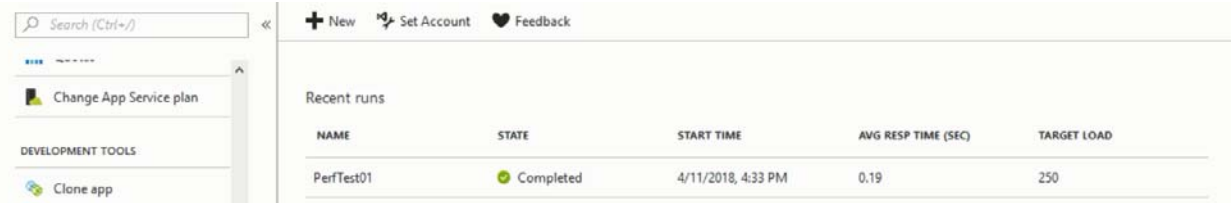


Figure 52 Performance test result

Figure 53 shows that the test request results are represented with a pie chart. The pie chart below shows that it is 100% successful and 0% of failure. The average response time is 0.19 and the request per second is 1.774.97.

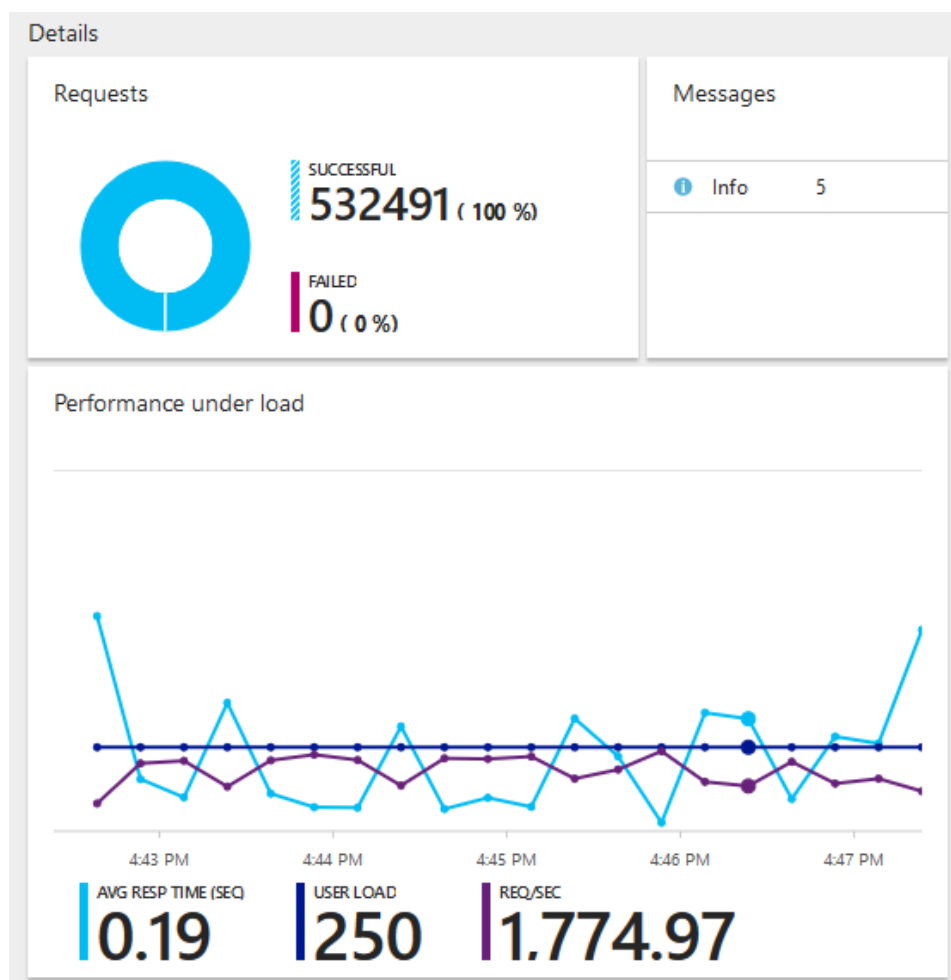


Figure 53 The details of the performance test

5.2 Unit Testing

5.2.1 Login

Test Case ID	Test Function	Test Case Description	Expected Result	Actual Result	Status (Pass/Fail)	Priority (High/Low)
T1	Login	1a. Enter Correct Email 1b. Enter Correct Password 2a. Enter Correct Email 2b. Enter Incorrect Password 3a. Enter Incorrect Email 3b. Enter Incorrect Password 4a. Enter Incorrect Email 4b. Enter Correct Password	1. Login Successfully 2. Login Failed 3. Login Failed 4. Login Failed	1. Login Successfully 2. Login Failed 3. Login Failed 4. Login Failed	Pass	High

5.2.2 Register Agent

Test Case ID	Test Function	Test Case Description	Expected Result	Actual Result	Status (Pass/Fail)	Priority (High/Low)
R1	Register Agent	1a. Enter Correct Email 1b. Enter Correct Password 2a. Enter Correct Email 2b. Enter Invalid Password 3a. Enter Invalid Email 3b. Enter Invalid Password 4a. Enter Invalid Email 4b. Enter Correct Password	1. Register Successfully 2. Register Failed 3. Register Failed 4. Register Failed	1. Register Successfully 2. Register Failed 3. Register Failed 4. Register Failed	Pass	High

5.2.3 Manage Customer

Test Case ID	Test Function	Test Case Description	Expected Result	Actual Result	Status (Pass/Fail)	Priority (High/Low)
C1	Create Customer	1. Enter Valid Details 2. Enter Invalid Details	1. Create customer details successfully 2. Create customer details unsuccessfully	1. Create customer details successfully 2. Create customer details unsuccessfully.	Pass	High
C2	Read Customer	1. Select particular customer details.	1. View customer details successfully.	1. View customer details successfully.	Pass	High
C3	Update Customer	1. Enter Valid Details 2. Enter Invalid Details	1. Update customer details successfully 2. Update customer details unsuccessfully	1. Update customer details successfully 2. Update customer details unsuccessfully.	Pass	High
C4	Delete customer	1. Select particular customer details.	1. View customer details successfully.	1. View customer details successfully.	Pass	High

5.2.4 Manage Booking

Test Case ID	Test Function	Test Case Description	Expected Result	Actual Result	Status (Pass/Fail)	Priority (High/Low)
B1	Create Booking	1. Enter Valid Details 2. Enter Invalid Details	1. Create booking successfully 2. Create booking unsuccessfully	1. Create booking successfully 2. Create booking unsuccessfully.	Pass	High
B2	Read Booking	1. Select particular booking details.	1. View booking details successfully.	1. View booking details successfully.	Pass	High

5.2.5 Manage Schedule

Test Case ID	Test Function	Test Case Description	Expected Result	Actual Result	Status (Pass/Fail)	Priority (High/Low)
SD1	Create Schedule	1. Enter Valid Details 2. Enter Invalid Details	1. Create schedule successfully 2. Create schedule unsuccessfully	1. Create schedule successfully 2. Create schedule unsuccessfully.	Pass	High
SD2	Read Schedule	1. Select particular schedule.	1. View schedule successfully.	1. View schedule successfully.	Pass	High
SD3	Update Schedule	1. Enter Valid Details 2. Enter Invalid Details	1. Update schedule successfully 2. Update schedule unsuccessfully	1. Update schedule successfully 2. Update schedule unsuccessfully.	Pass	High
SD4	Delete Schedule	1. Select particular schedule.	1. View schedule successfully.	1. View schedule successfully.	Pass	High

5.2.6 Manage Ship

Test Case ID	Test Function	Test Case Description	Expected Result	Actual Result	Status (Pass/Fail)	Priority (High/Low)
S1	Create Ship	1. Enter Valid Details 2. Enter Invalid Details	1. Create ship details successfully 2. Create ship details unsuccessfully	1. Create ship details successfully 2. Create ship details unsuccessfully.	Pass	High
S2	Read Ship	1. Select particular ship details.	1. View ship details successfully.	1. View ship details successfully.	Pass	High
S3	Update Ship	1. Enter Valid Details 2. Enter Invalid Details	1. Update ship details successfully 2. Update ship details unsuccessfully	1. Update ship details successfully 2. Update ship details unsuccessfully.	Pass	High
S4	Delete Ship	1. Select particular ship details.	1. View ship details successfully.	1. View ship details successfully.	Pass	High

Chapter 6: CONCLUSION

In conclusion, Maersk Line - Container Management System (CMS) has designed, developed and deployed successfully to Microsoft Azure Cloud Platform. The web application is designed with the help of Visual Studio, where ASP.Net MVC framework will be used. The web application used is C# programming language. The web application connects to SQL server and SQL database that are hosted in Microsoft Azure. Besides, the resource group, geo-replication, traffic manager, auto scaling and performance test is completed after the project is developed completely via Visual Studio. Unit testing is conducted before the publishing of web application to check for defects and deficiencies. When errors appear, developers are required to fix them before publishing to Azure Cloud.

With the help of Microsoft Azure, the business satisfaction has met to aid in the growth of Maersk Line Business and the daily support container management. It also able to reduce the cost of the Maersk Line Business and have an efficient way to manage the Maersk Line Logistics.

Chapter 7: REFERENCE

Azure, M., 2018. *What are Azure SQL Database service tiers?*. [Online]
Available at: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tiers>
[Accessed 21 February 2018].

Microsoft, 2018. *ASP.NET MVC Overview*. [Online]
Available at: [https://msdn.microsoft.com/en-us/library/dd381412\(v=vs.108\).aspx](https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx)
[Accessed 10 February 2018].

Petrossian, T., 2014. *Azure SQL Database Standard Geo-Replication*. [Online]
Available at: <https://azure.microsoft.com/en-us/blog/azure-sql-database-standard-geo-replication/>
[Accessed 20 February 2018].

Chapter 8: APPENDIX

1. Source Code (GitHub repository)

- a. <https://github.com/magpie0511/MaerskLineWebsite>

2. Azure & System Demonstration Video

<https://web.microsoftstream.com/video/d97705c5-d578-4aae-88e7-cd9ad3e8d6ff>

3. Azure Web App URL

- a. Traffic Manager: <http://maerskline-cms-tp035869.trafficmanager.net>
- b. Southeast Asia: <https://tp035869-maerskline-sea.azurewebsites.net>
- c. North Central US: <https://tp035869-maerskline-ncus.azurewebsites.net>

4. Sample User Credential for Testing

a. Admin Account

- Email: admin@maersk.com
- Password: Admin123.

b. Agent Account

- Email: agent@maersk.com
- Password: Agent123.