



Designing and Developing

Applications on the Cloud

CT071-3-3-DDAC

Lecturer : Dr. Kalai Anand A/L Ratnam

Student Name : Wan Cai Ling

TP Number : TP035362

Intake : UC3F1706SE

Title : Maersk Line Container Management
System

Due Date : 13th April 2018

Table of Contents

1.0 Introduction.....	5
1.1 Background	5
1.2 Aim.....	5
1.3 Objectives.....	5
1.4 Scope	6
1.5 Non-functional Requirement Specification.....	6
1.6 Functional Requirement Specification	7
2.0 Project Plan	8
2.1 Work Breakdown Structure.....	8
2.2 Gantt Chart	9
3.0 Design	12
3.1 Cloud Architecture Diagram	12
3.2 Design Consideration	13
3.3 Modelling	13
3.3.1 Use Case Diagram	13
3.3.2 Use Case Specification	14
3.3.3 Sequence Diagram	24
3.3.4 Class Diagram	29
4.0 Implementation	30
4.1 ASP.NET Web Application	30
4.2 Azure Resource Group	31
4.2.1 Resource group for web-application service	32
4.2.2 Resource group for SQL Database	33
4.2.3 Resource group for traffic manager.....	34
4.3 Azure SQL Server	35

4.3.1 Setup SQL Server	35
4.4 Azure SQL Database.....	36
4.4.1 Setup Primary SQL Database	37
4.4.2 Setup Secondary Database.....	38
4.5 Configure Failover Policy	41
4.6 Firewalls and Virtual Networks	43
4.7 Azure Web Application Services	45
4.8 Deploy ASP.NET Web Application.....	49
4.9 Azure Traffic Manager.....	54
4.9.1 Setup Traffic Manager Profile	54
4.9.2 Setup Endpoint	55
5.0 Testing.....	56
5.1 Unit Testing.....	56
5.1.1 Login.....	56
5.1.2 Booking Ship Schedule	57
5.1.3 View Customer	58
5.1.4 Register Customer	58
5.1.5 View Ship	59
5.1.6 Add Ship	59
5.1.7 View Schedule	60
5.1.8 Add Schedule.....	61
5.1.9 Edit Schedule	62
5.1.10 View Booking.....	62
5.1.11 Register Agent	63
5.1.12 Log Off	64
5.2 Performance Testing	65

6.0 Conclusion	68
7.0 References.....	69
8.0 Appendix.....	70

1.0 Introduction

1.1 Background

Maersk Line is the world's largest container shipping company founded in the year of 1928, operating over 100 countries and transporting goods globally. The company having customers through 374 offices in 116 countries. Besides, Maersk Lines operates more than 600 vessels and has a capacity of 2.6 million TEU. Maersk Line has realized that the volume of most of the goods for shipping has grown to full capacity and hard to be managed. The company was looking forwards a solution which can support further business growth and increase organizational flexibility. The company has decided to consolidate all the data centres and server rooms operating worldwide onto visualized platform.

To achieve the needs of the company, the company has decided to use cloud as the solution. Microsoft Azure was chosen to host some its IT environment. The company has planned to expand the scope of the relationship by altering its IT setup based on Microsoft Azure, which may change the current company desktop environment into a better container management.

1.2 Aim

To develop quality a cloud-based application that reduces overall supply chain costs and provide an efficient way to manage logistics of Maersk Line.

1.3 Objectives

Develop a container management web application and publish it to Microsoft Azure Cloud services, which allows admin to manage the Maersk cargo and shipping operations. The web application implements with add, update and view functions to manage the entire shipping process operations. The web application and database of the container management system are hosted to cloud environment with failover and performance management which may lead to high performance system.

1.4 Scope

The scope of the project consists of design and development of the web application based on the requirements set by Maersk Line and deploy the system to cloud environment through Microsoft Azure, letting users from different geographical areas to access the system easily.

1.5 Non-functional Requirement Specification

Non-functional requirements	Descriptions
Provisioning	To provision new application which is container management system (CMS) to the Microsoft Azure platform.
Monitoring	To monitor the application for problem identification and troubleshooting.
Scalability	To make sure that the application meets the requirements.
Availability	To ensure that the web application can available online always.
Maintainability	To upgrade the application and perform maintenance jobs in the condition where users are concurrently using the application.

Table 1: Non-functional requirement specification of CMS system

1.6 Functional Requirement Specification

Maersk Line CMS is a cloud-based application that can run on various web browsers which includes Google Chrome, Mozilla Firefox and so forth. The system users will be agent and admin they can able to perform the system functions as shown in table below:

User	Functional Requirement
Agent	<ul style="list-style-type: none">• Login• View Customer• Register Customer• Book Shipping Schedule
Agent	<ul style="list-style-type: none">• Login• View Ship• Add Ship• View Schedule• Edit Schedule• Add Schedule• View Booking• Register Agent

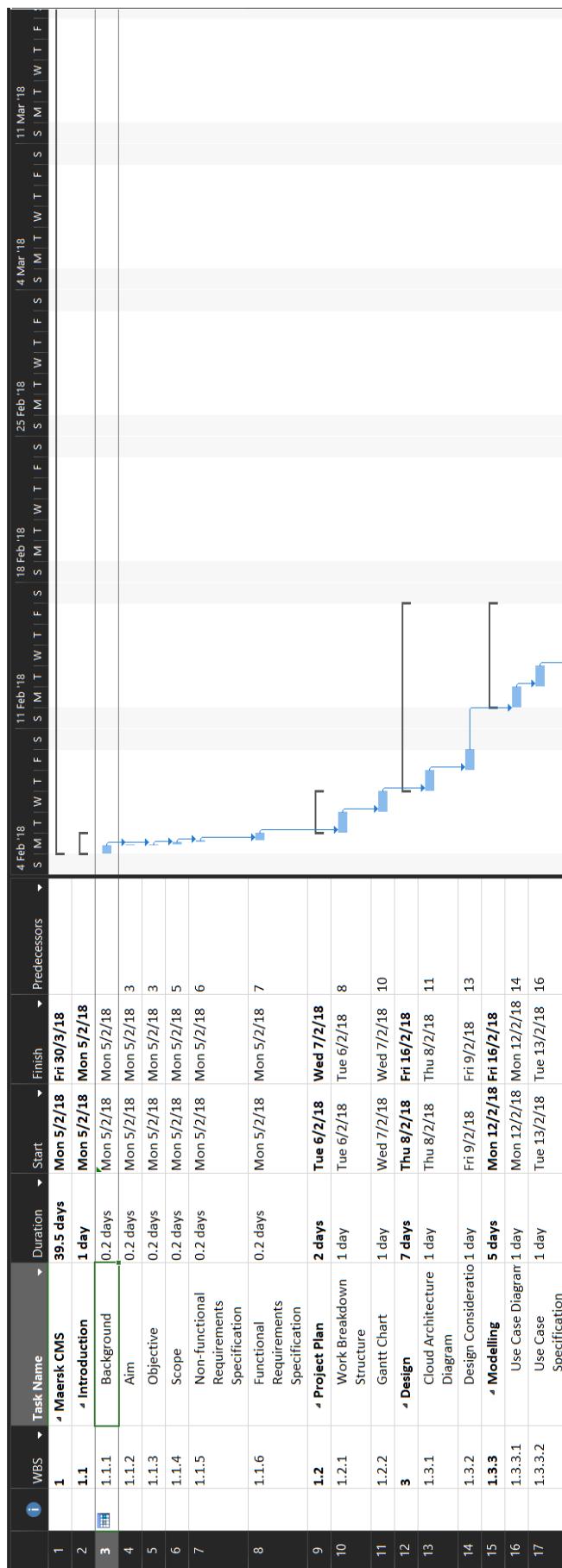
Table 2: Functional Requirement Specification

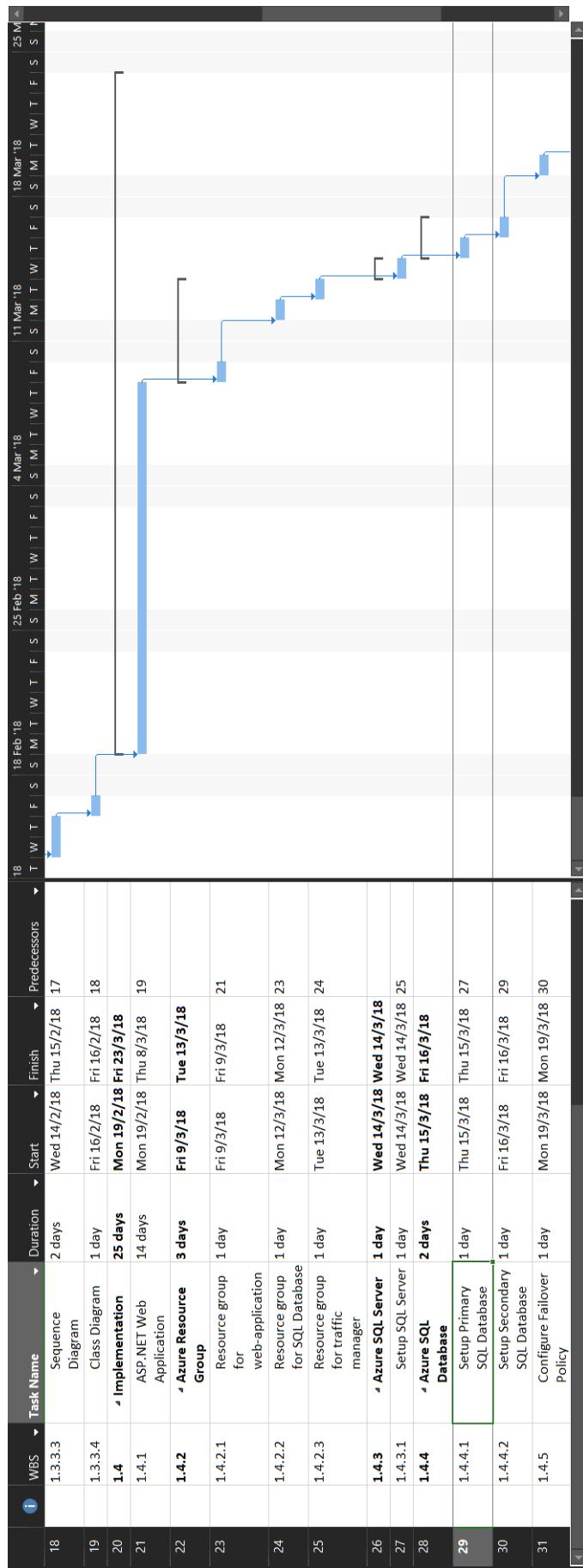
2.0 Project Plan

2.1 Work Breakdown Structure

WBS	Task Name	Duration
1	Maersk CMS	39.5 days
1.1	Introduction	1 day
1.1.1	Background	0.2 days
1.1.2	Aim	0.2 days
1.1.3	Objective	0.2 days
1.1.4	Scope	0.2 days
1.1.5	Non-functional Requirements Specification	0.2 days
1.1.6	Functional Requirements Specification	0.2 days
1.2	Project Plan	2 days
1.2.1	Work Breakdown Structure	1 day
1.2.2	Gantt Chart	1 day
3	Design	7 days
1.3.1	Cloud Architecture Diagram	1 day
1.3.2	Design Consideration	1 day
1.3.3	Modelling	5 days
1.3.3.1	Use Case Diagram	1 day
1.3.3.2	Use Case Specification	1 day
1.3.3.3	Sequence Diagram	2 days
1.3.3.4	Class Diagram	1 day
1.4	Implementation	25 days
1.4.1	ASP.NET Web Application	14 days
1.4.2	Azure Resource Group	3 days
1.4.2.1	Resource group for web-application service	1 day
1.4.2.2	Resource group for SQL Database	1 day
1.4.2.3	Resource group for traffic manager	1 day
1.4.3	Azure SQL Server	1 day
1.4.3.1	Setup SQL Server	1 day
1.4.4	Azure SQL Database	2 days
1.4.4.1	Setup Primary SQL Database	1 day
1.4.4.2	Setup Secondary SQL Database	1 day
1.4.5	Configure Failover Policy	1 day
1.4.6	Firewalls and Virtual Networks	1 day
1.4.7	Azure Web Application Services	1 day
1.4.8	Deploy ASP.NET Web Application	1 day
1.4.9	Azure Traffic Manager	1 day
1.5	Testing	2 days
1.5.1	Unit Testing	1 day
1.5.2	Performance Testing	1 day
1.6	Conclusion	1 day
1.7	References	1 day
1.8	Appendix	0.5 days

2.2 Gantt Chart







3.0 Design

3.1 Cloud Architecture Diagram

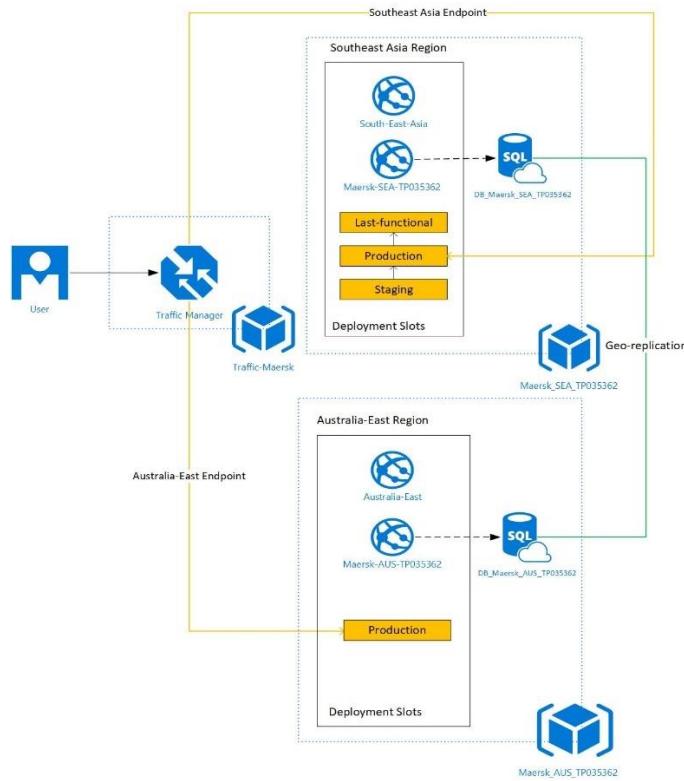


Figure 1: Maersk Line Container Management System Cloud Architecture Diagram

The figure above shows the cloud architecture diagram for the deployment of Maersk Line Container Management System to Microsoft Azure cloud platform. The design consists of multi-region which the web application will be deployed to Southeast Asia and Australia-East regions, where the primary is located in Southeast Asia and the secondary is located at Australia-East. In each of the region has its own app service, app service plan, SQL server and SQL database which used to support the Maersk Line CMS system deployment to cloud platform. The geo-replication service is applied to the databases in both regions where the primary database is located in Southeast Asia and the secondary database is located in Australia-East region. This is used to manage the failover situation of database by copying the data from primary database to secondary database in order to prevent any data loss that may affect the data availability. Each region has its endpoint of web apps which managed by traffic manager to improves the web application performance by redirecting the client request to the nearest web app service.

3.2 Design Consideration

There are several assumptions that need to be considered before the system design begins. The aim of Maersk Line is to reduce overall supply chain costs and provide efficient way to manage the logistics through utilizing the cloud-based application. Hence, the development team decides to host the Container Management system to cloud platform with the main focus in Southeast Asia region where the primary target users are located. On the other hand, by focusing on deploying Southeast Asia do not require much expenses on the hosting fees. Besides, as Maersk Line run its business globally, the development team also planning to test the system in Australia-East by deploying the web application to that region as well. If the testing is success, the company will feel confident to deploy the web applications in different countries and then slowly expand its business efficiently with minimal spending, leading a profitable advantage to the Maersk Line company.

3.3 Modelling

3.3.1 Use Case Diagram

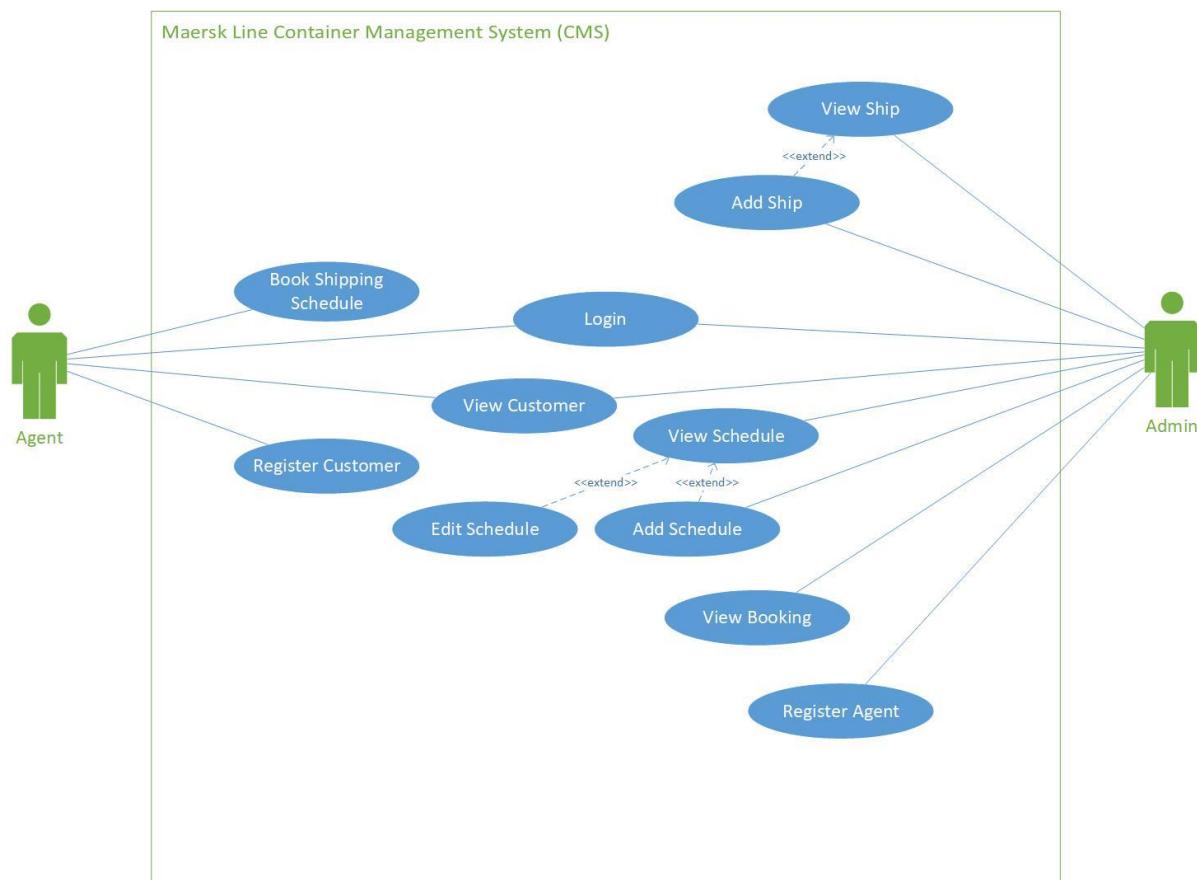


Figure 2: Maersk Line Container Management System Use Case Diagram

3.3.2 Use Case Specification

3.3.2.1 Login

Use Case	Login
Summary	User login into the system to perform tasks that offered by the system according to their user privilege.
Dependency	-
Actors	Agent, Admin
Preconditions	The particular user has been registered through the system
Descriptive of Main Sequence	<ol style="list-style-type: none"> 1. User click Login. 2. System display login page. 3. User provide their email and password. 4. System verifies the email and password provided by the user. 5. System grant the access permission and direct user to home page with relative function according to the type of user account.
Descriptive of Alternative Sequence	<p>3a. If the email and password entered by user is invalid the system will display error message to acknowledge user to re-enter the respective information.</p> <p>3b. If user do not fully fill in all the login criteria, the system will display error message to acknowledge user to fill in all the required information.</p>
Postcondition	User login successful and may perform tasks in the system.

Table 3: Login use case specification

3.3.2.2 Book Shipping Schedule

Use Case	Book Shipping Schedule
Summary	Agent book schedule for shipping purpose.
Dependency	-
Actors	Agent
Preconditions	User logged into the system.
Descriptive of Main Sequence	<ol style="list-style-type: none"> 1. Select Book Schedule. 2. System retrieve data from database. 3. System browse booking index page. 4. Select the ship schedule to be booked. 5. System retrieve data of the selected ship schedule from database. 6. System display make booking page. 7. Choose customer who assigned the booking. 8. Enter the container type for the booking. 9. Enter the number of bay used. 10. Click on booking button. 11. System verifies the input provided by user. 12. System save information to database. 13. System display “Book successful” message.
Descriptive of Alternative Sequence	<p>7a. If user do not choose the customer and click on book button, the system will display error message to acknowledge user to choose customer.</p> <p>8a. If the input for the container type is invalid, the system will display error</p>

	<p>message to acknowledge user to re-enter the respective information.</p> <p>8b. If user do not enter any value for the container type and click on booking button, the system will display error message to acknowledge user to enter the required information.</p> <p>9a. If the input for the number of bay used is invalid, the system will display error message to acknowledge user to re-enter the respective information.</p> <p>9b. If user do not enter any value for the number of bay used and click on booking button, the system will display error message to acknowledge user to enter the required information.</p> <p>9c. If the number of bay used is larger than the number of remaining cargo size of the ship, the system will display error message to acknowledge user regarding the issue and not allow user to proceed the booking.</p>
Postcondition	User successfully make booking on the selected ship schedule.

Table 4: Book Shipping Schedule use case specification

3.3.2.3 View Customer

Use Case	View Customer
Summary	View registered customer information.
Dependency	-
Actors	Agent, Admin
Preconditions	User logged into the system.
Descriptive of Main Sequence	<ol style="list-style-type: none"> 1. Select View Customer. 2. System retrieve data from database. 3. System browse view customer index page. 4. User view customer information that displayed in the view customer page.
Descriptive of Alternative Sequence	-
Postcondition	User successfully view registered customer information.

Table 5: View Customer use case specification

3.3.2.4 Register Customer

Use Case	Register Customer
Summary	Register customer with relevant customer's information for allow the customer can perform booking schedule.
Dependency	-
Actors	Agent
Preconditions	User logged into the system.
Descriptive of Main Sequence	<ol style="list-style-type: none"> 1. User select Register Customer. 2. System display Register Customer page. 3. User enter all the required information. 4. System verifies the input provided by the user.

	<p>5. System save information to database.</p> <p>6. System direct user to view customer page to let user view at the updated registered customer information.</p>
Descriptive of Alternative Sequence	<p>3a. If user do not enter all the required information with valid input, system will display error message and user have to re-enter the required information with valid input.</p> <p>3b. If user did not enter all relevant information that required to be filled in, system will display error message for to notify user to fill in all the information completely.</p>
Postcondition	User successfully registered new customer.

Table 6: Register customer use case specification

3.3.2.5 View Ship

Use Case	View Ship
Summary	View ship information.
Dependency	<<extend>> Add Ship
Actors	Admin
Preconditions	User logged into the system.
Descriptive of Main Sequence	<p>1. Select View Ship.</p> <p>2. System retrieve data from database.</p> <p>3. System browse ship index page.</p> <p>4. User view ship information that displayed in the ship index page.</p> <p>If user choose to add ship:</p> <p>5. Extend Add Ship Use Case.</p>
Descriptive of Alternative Sequence	-
Postcondition	User successfully view ship information.

Table 7: View ship use case specification

3.3.2.6 Add Ship

Use Case	Add Ship
Summary	Add new ship with its relevant information.
Dependency	-
Actors	Admin
Preconditions	User logged into the system.
Descriptive of Main Sequence	<ol style="list-style-type: none"> 1. User select Add Ship. 2. System display Add Ship page. 3. User enter all the required information. 4. System verifies the input provided by the user. 5. System save information to database. 6. System direct user to ship index page to let user view at the ship information that has added and updated.
Descriptive of Alternative Sequence	<p>3a. If user do not enter all the required information with valid input, system will display error message and user have to re-enter the required information with valid input.</p> <p>3b. If user did not enter all relevant information that required to be filled in, system will display error message for to notify user to fill in all the information completely.</p>
Postcondition	User successfully add ship with its relevant information.

Table 8: Add ship use case specification

3.3.2.7 View Schedule

Use Case	View Schedule
Summary	View schedule information.
Dependency	<<extend>> Add Schedule <<extend>> Edit Schedule
Actors	Admin
Preconditions	User logged into the system.
Descriptive of Main Sequence	<ol style="list-style-type: none"> 1. Select View Schedule. 2. System retrieve data from database. 3. System browse schedule index page. 4. User view schedule information that displayed in the schedule index page. <p>If user choose to add schedule:</p> <ol style="list-style-type: none"> 5. Extend Add Schedule Use Case. <p>If user choose to edit schedule:</p> <ol style="list-style-type: none"> 6. Click on edit link. 7. System browse edit schedule page. 8. User edit schedule information. 9. Click on edit button. 10. System validate the input provided by user. 11. System update user information in database. 12. System direct user to schedule index page to let user view at the update schedule information that has edited.
Descriptive of Alternative Sequence	8a. If user do not fully fill in the required information, system will display error message for to notify user to fill in the information completely.
Postcondition	<ol style="list-style-type: none"> 1. User successfully view schedule information.

	2. User successfully update the schedule information.
--	---

Table 9: View schedule use case specification

3.3.2.8 Add Schedule

Use Case	Add Schedule
Summary	Add new ship schedule with its relevant information.
Dependency	-
Actors	Admin
Preconditions	User logged into the system.
Descriptive of Main Sequence	<ol style="list-style-type: none"> 1. User select Add Schedule. 2. System display Add Schedule page. 3. User enter all the required information. 4. System verifies the input provided by the user. 5. System save information to database. 6. System direct user to schedule index page to let user view at the schedule information that has added and updated.
Descriptive of Alternative Sequence	<p>3a. If user do not enter the required information with valid input, system will display error message and user have to re-enter the required information with valid input.</p> <p>3b. If user did not enter all relevant information that required to be filled in, system will display error message for to notify user to fill in all the information completely.</p>

Postcondition	User successfully add ship schedule with its relevant information.
----------------------	--

Table 10: Add schedule use case specification

3.3.2.9 View Booking

Use Case	View Booking
Summary	View the booking information.
Dependency	-
Actors	Admin
Preconditions	User logged into the system.
Descriptive of Main Sequence	<ol style="list-style-type: none"> 1. Select View Booking. 2. System retrieve data from database. 3. System browse view booking page. 4. User view booking information that displayed in the view booking page.
Descriptive of Alternative Sequence	-
Postcondition	User successfully view booking information.

Table 11: View booking use case specification

3.3.2.10 Registered Agent

Use Case	Register Agent
Summary	Register agent with relevant agent's information for allowing the agent can perform agent's tasks using agent account.
Dependency	-
Actors	Admin
Preconditions	User logged into the system.
Descriptive of Main Sequence	<ol style="list-style-type: none"> 1. User select Register Agent. 2. System display Register Agent page. 3. User enter all the required information. 4. System verifies the input provided by the user.

	<p>5. System save information to database.</p> <p>6. System direct user to home index page.</p>
Descriptive of Alternative Sequence	<p>3a. If user do not enter all the required information with valid input, system will display error message and user have to re-enter the required information with valid input.</p> <p>3b. If user did not enter all relevant information that required to be filled in, system will display error message for to notify user to fill in all the information completely.</p>
Postcondition	User successfully registered new agent.

Table 12: Register agent use case specification

3.3.3 Sequence Diagram

3.3.3.1 Login

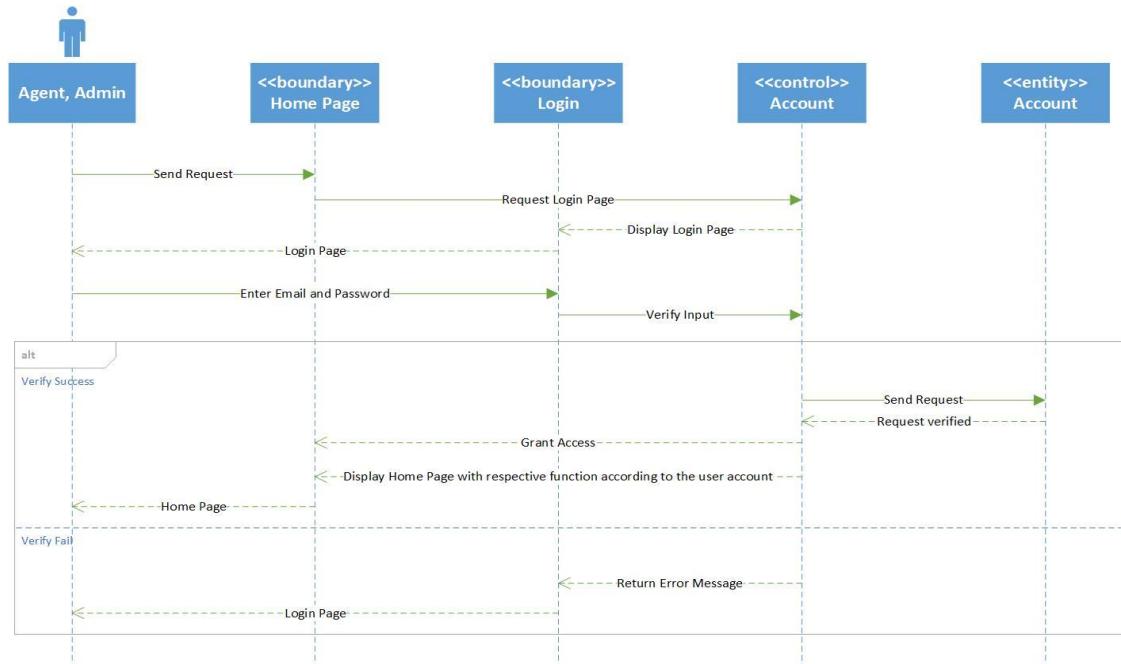


Figure 3: Login Sequence Diagram

3.3.3.2 Booking Shipping Schedule

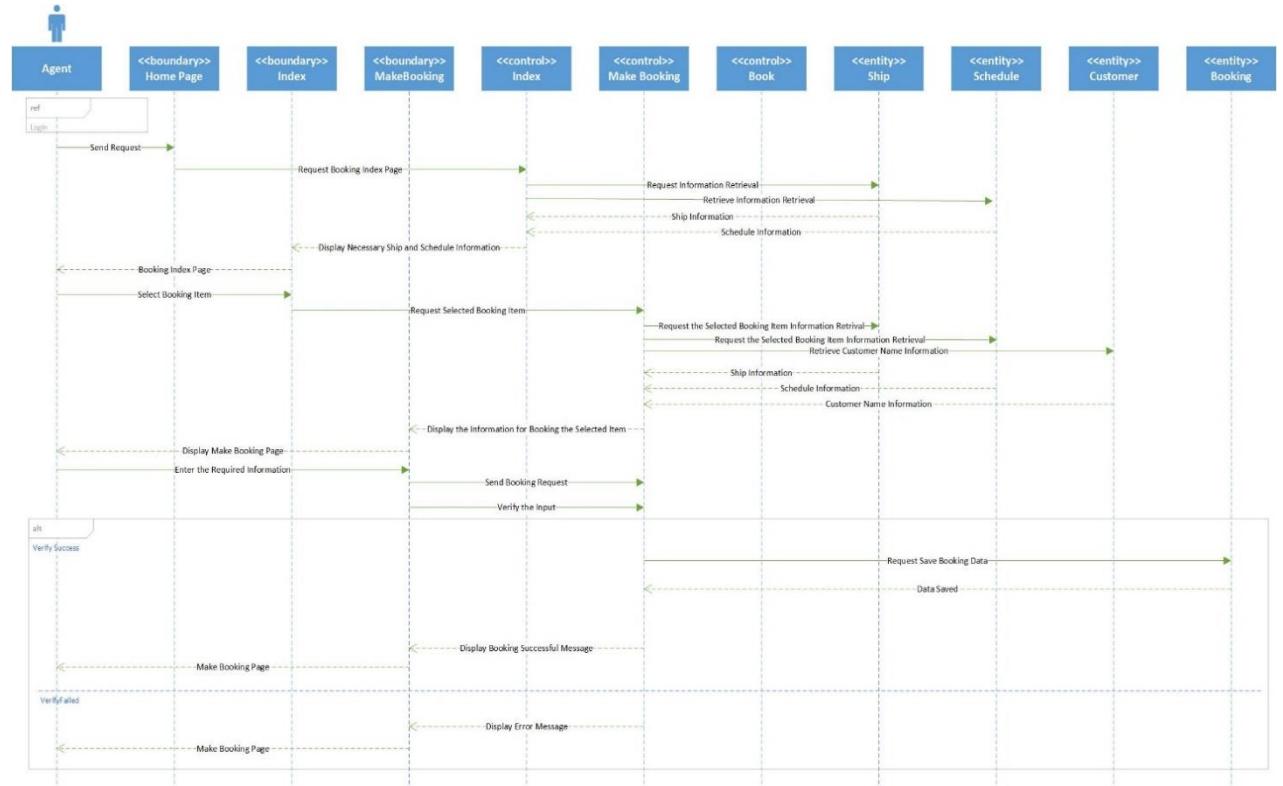


Figure 4: Booking Shipping Schedule Sequence Diagram

3.3.3.3 View Customer

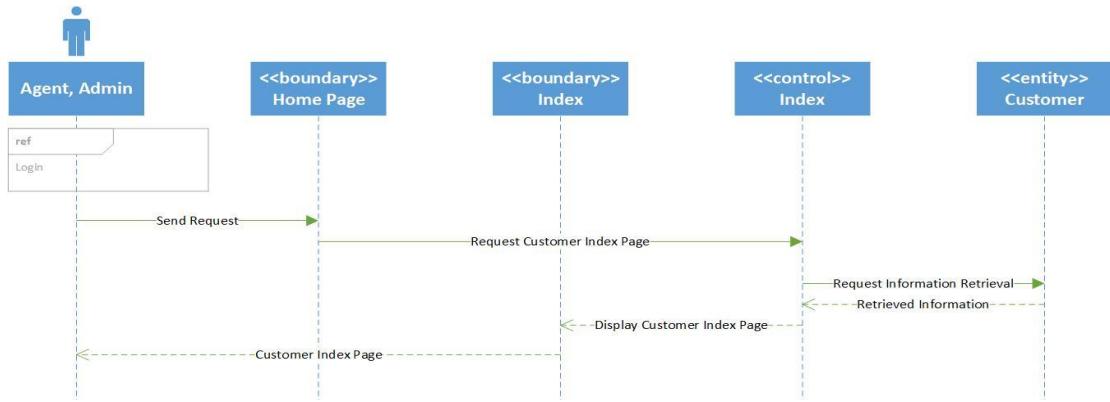


Figure 5: View Customer Sequence Diagram

3.3.3.4 Register Customer

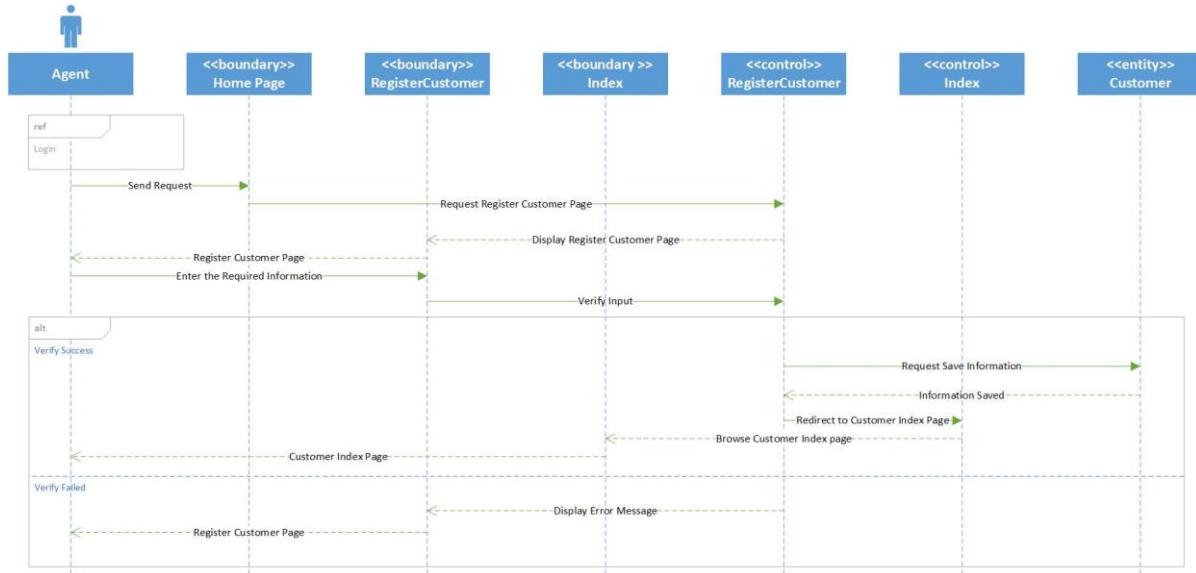


Figure 6: Register Customer Sequence Diagram

3.3.3.5 View Ship

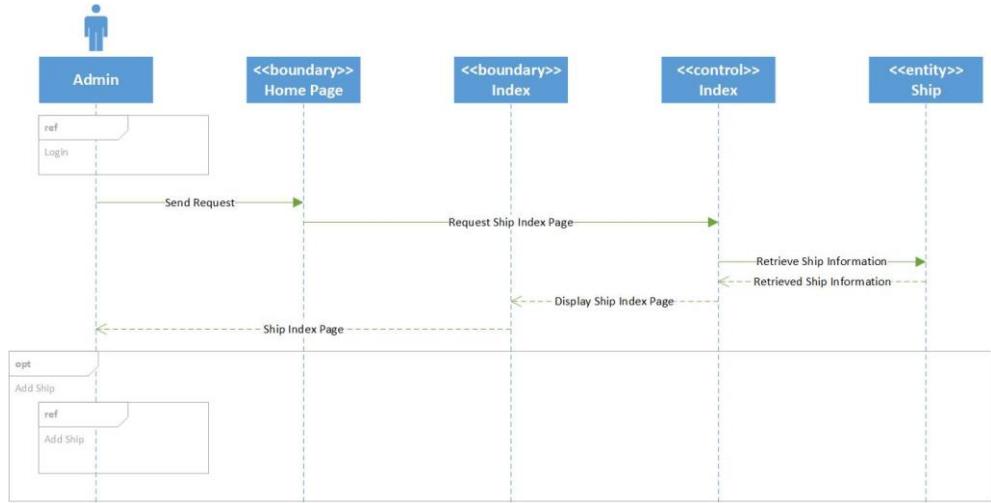


Figure 7: View Ship Sequence Diagram

3.3.3.6 Add Ship

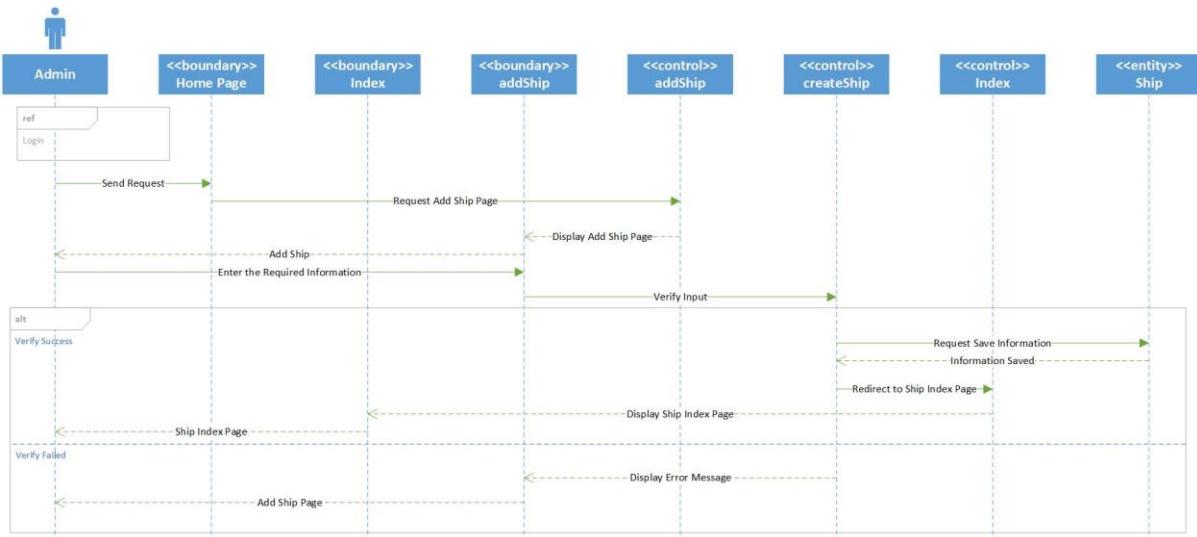


Figure 8: Add Ship Sequence Diagram

3.3.3.7 View Schedule

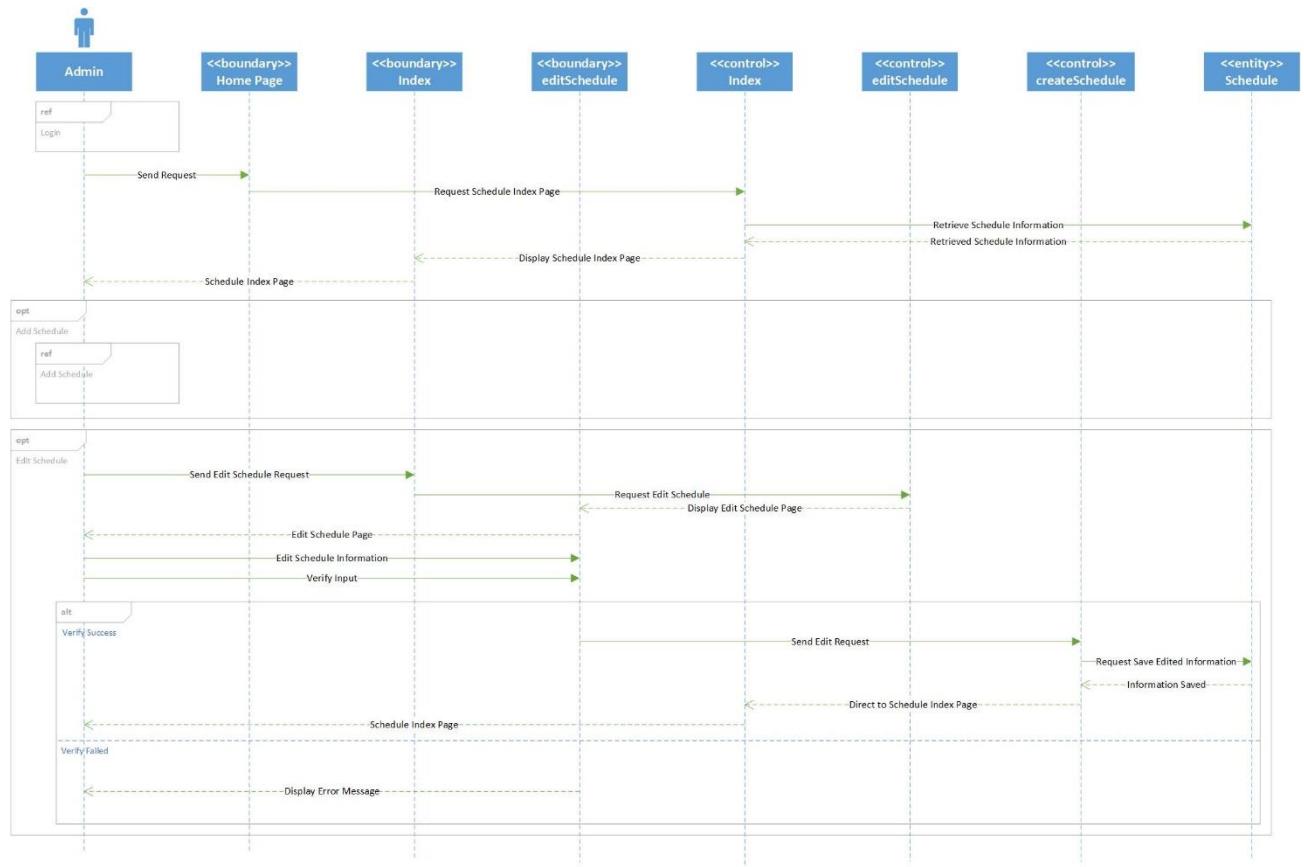


Figure 9: View Schedule Sequence Diagram

3.3.3.8 Add Schedule

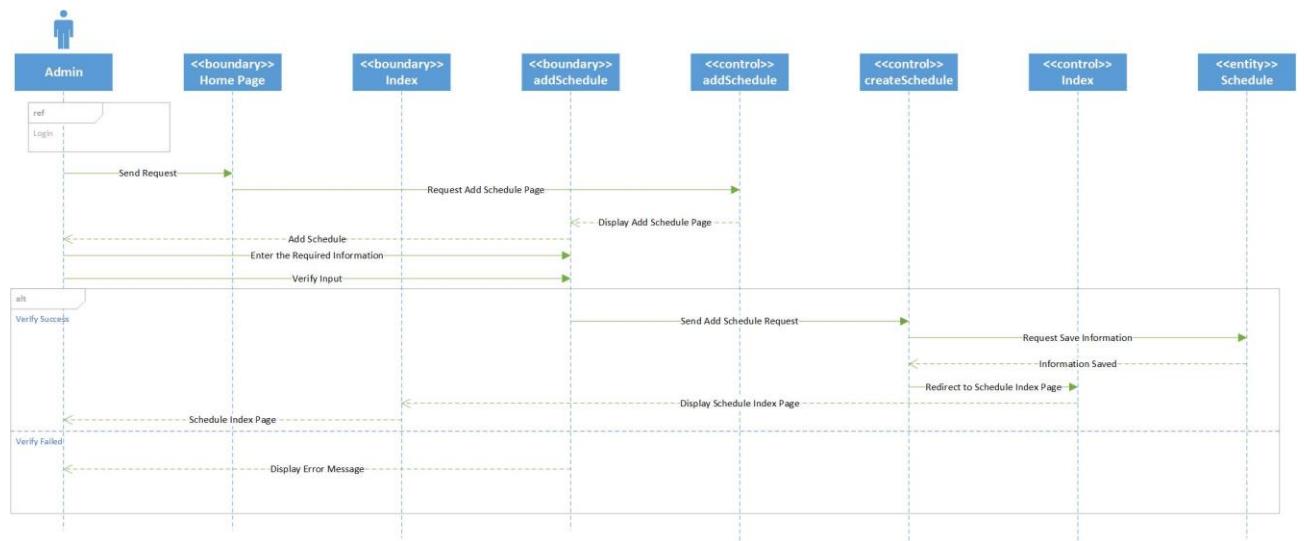


Figure 10: Add Schedule Sequence Diagram

3.3.3.9 View Booking

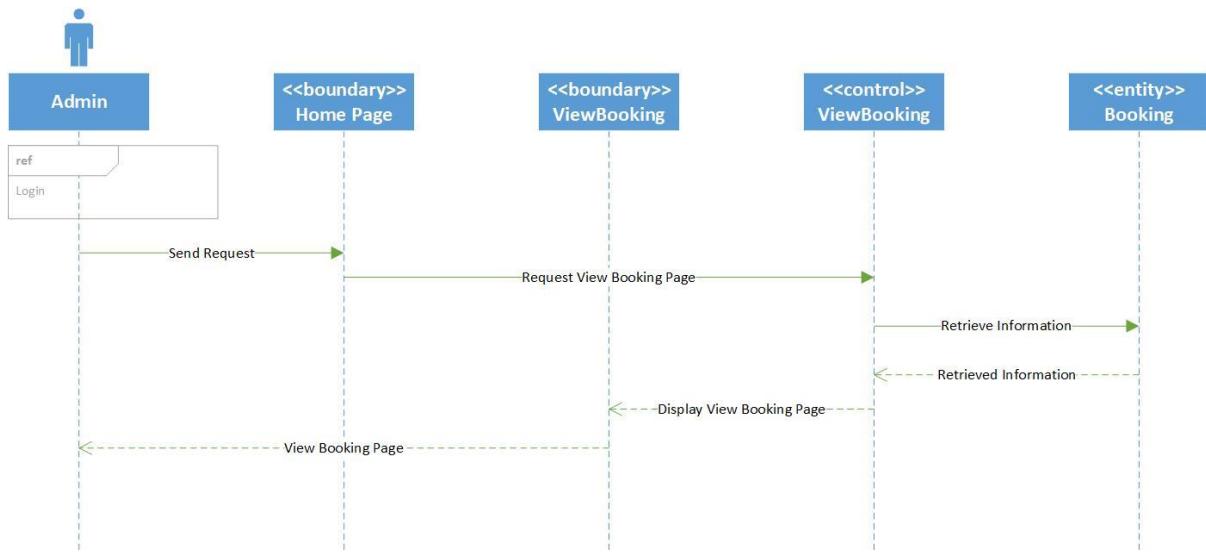


Figure 11: View Booking Sequence Diagram

3.3.3.10 Register Agent

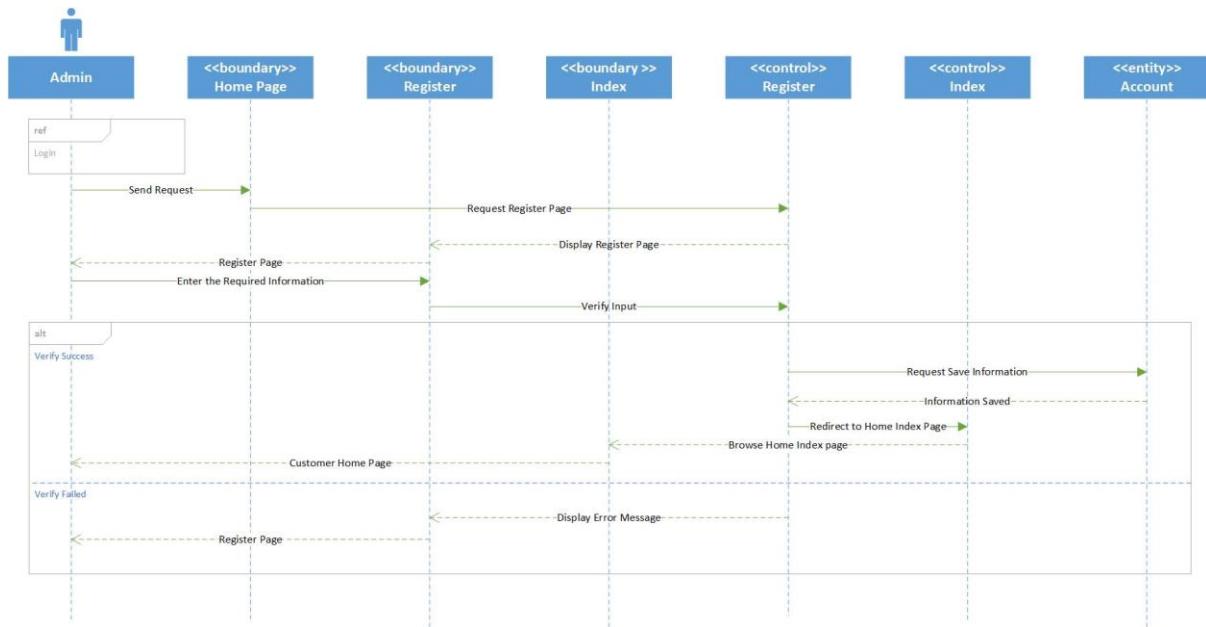


Figure 12: Register Agent Sequence Diagram

3.3.4 Class Diagram

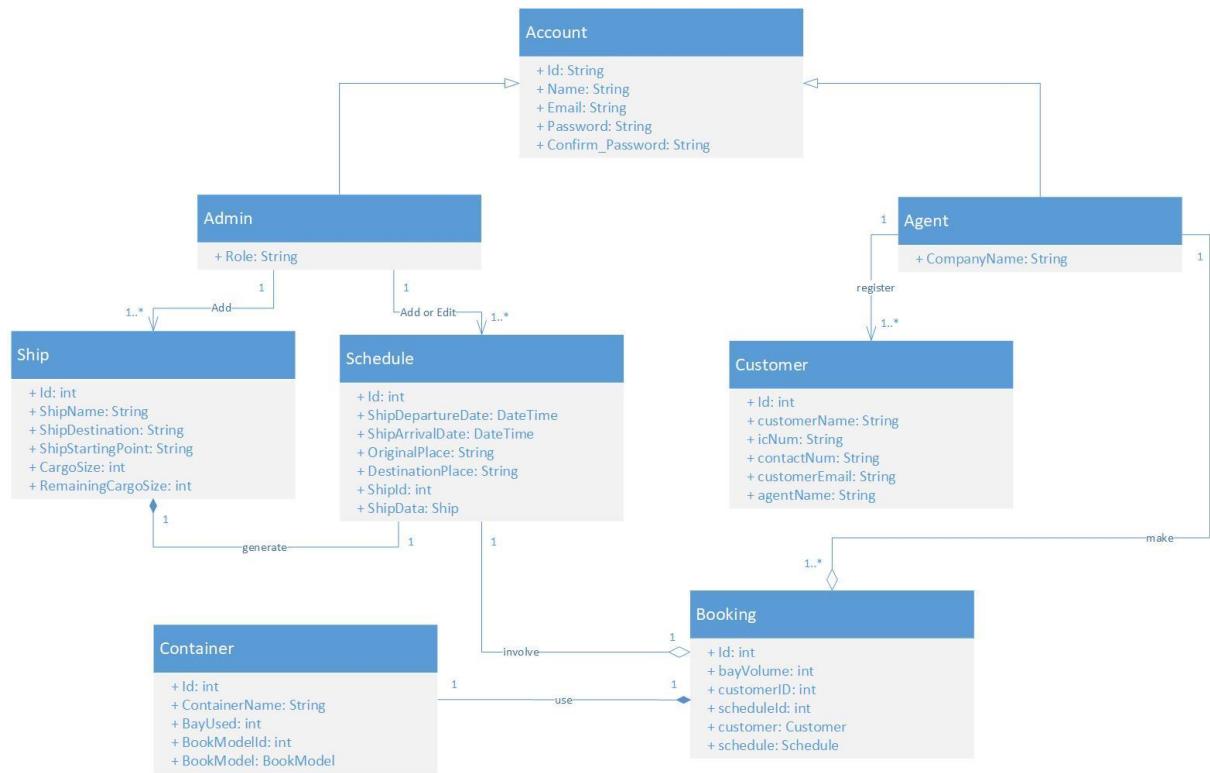


Figure 13: Maersk Line Container Management System Class Diagram

4.0 Implementation

4.1 ASP.NET Web Application

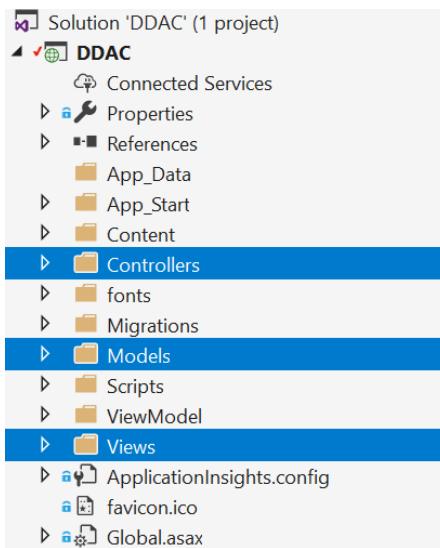


Figure 14: ASP.NET MVC model

The Maersk Line Container Management System (CMS) is developed using ASP.NET MVC development framework. It is a framework that implements the view-controller-model pattern in which it consists of 3 main components which are view, controller and model. View interacts with user interface (UI); controller represent the components that handle user interaction, work with the model, and eventually select a view to render that displays user interface (UI); model is responsible in interacting with database. MVC framework support rapid and parallel development, it also reduces code duplication because the framework separated the data and business logic allowing the modification of code without affecting the entire model which process data and the whole system architecture. Code-first approach is being used which focus on creating classes for the domain entity rather than design the database first through writing codes (Kayal, 2014). Code first approach is being used as the system being developed is a brand-new system with no existing database and the developer can able to obtain full control over the database design through coding.

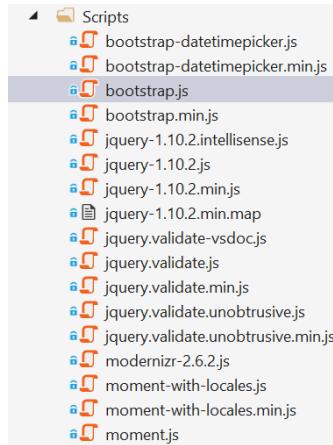


Figure 15: Bootstrap Scripts File

At the same time, web responsive design is implemented using bootstrap which is a front-end web framework. This allow the system to be responsive by automatically adjust the size of the system appearance accordingly based on various device.

4.2 Azure Resource Group

NAME	SUBSCRIPTION	LOCATION
DB_Maersk_AUS_TP035362	Azure for Students	Australia East
DB_Maersk_SEA_TP035362	Azure for Students	Southeast Asia
Maersk_AUS_TP035362	Azure for Students	Australia East
Maersk_SEA_TP035362	Azure for Students	Southeast Asia

Figure 16: Resource groups for the deploy of Maersk Line CMS system through Microsoft Azure

4.2.1 Resource group for web-application service

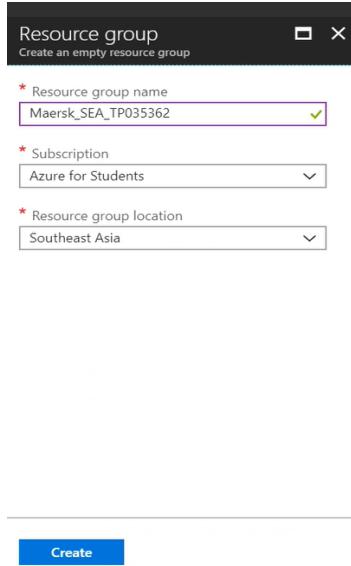


Figure 17: Web application service resource group for Southeast Asia

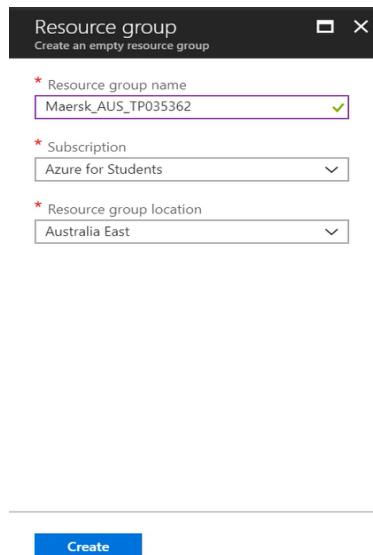


Figure 18: Web application service resource group for Australia-East

The resource groups are created to contain the related resources or services for the web application in the Azure platform. Two resource groups with different names represent the deployment of web application for 2 different regions which are Southeast Asia and Australia East. The resource group that deploys to Southeast Asia acts as the primary resource group for the web application, whereas the resource group that deploys to Australia East acts as the secondary resource group. Southeast Asia is chosen as the primary resource group as the primary target users of the Maersk Line CMS system are coming from this region.

4.2.2 Resource group for SQL Database

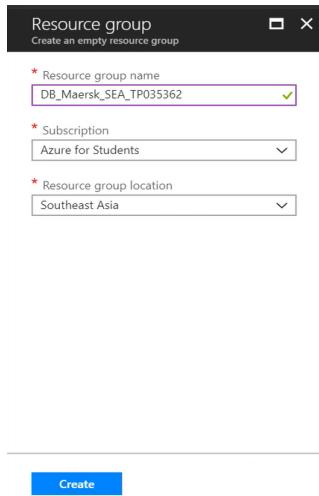


Figure 19: Database resource group for Southeast Asia

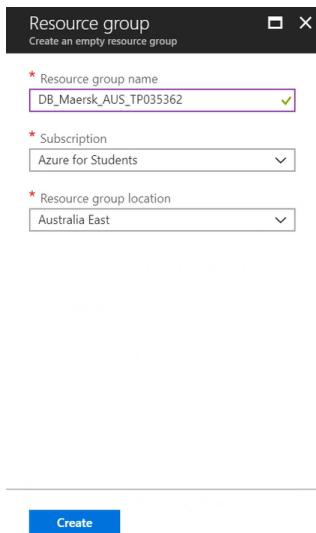


Figure 20: Database resource group for Australia-East

Resource group for database is created to hold the database related services for the database performance such as SQL Server and SQL database. Two resource groups are created for two different regions which are Southeast Asia and Australia East. The primary database is set in the Southeast Asia, whereas the secondary database is set in the Australia East. The primary and secondary database share the same source which is from the database of Maersk Line CMS system. Meaning that the content of both primary and secondary resource group will be the same, any update of the primary database will be received immediately by secondary database. Thus, if there is failure in primary database, the secondary database can be used to support the primary database to ensure data availability.

4.2.3 Resource group for traffic manager

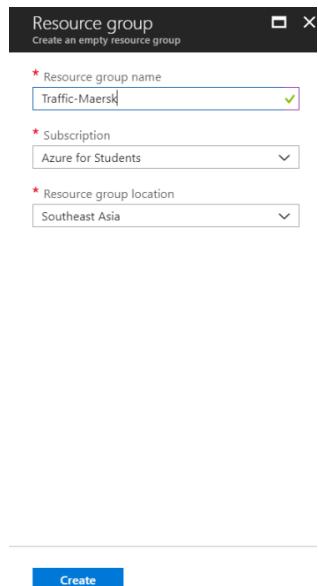


Figure 21: Traffic Manager resource group

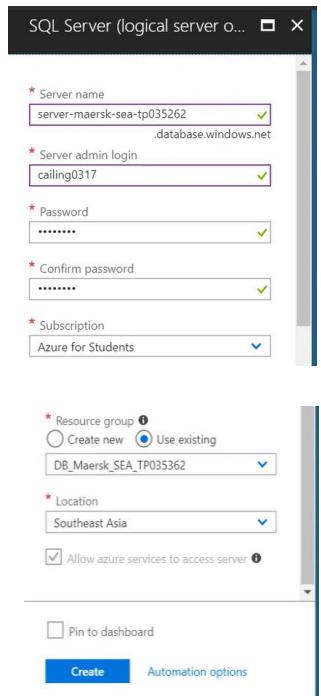
Create a resource group for traffic manager in order to store all the service that support with the traffic manager operation. The resource group name is set as “Traffic-Maersk”. The resource group location set in Southeast Asia as is it the region where the primary web app is hosted.

4.3 Azure SQL Server

SQL servers ALL SUBSCRIPTIONS		
 server-maersk-sea-tp035262	SQL server	Southeast Asia
 server-maersk-aus-tp035362	SQL server	Australia East

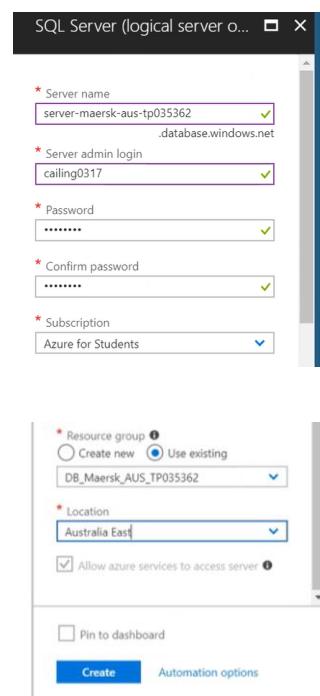
Figure 22: SQL server for Southeast Asia and Australia-East Regions

4.3.1 Setup SQL Server



The screenshot shows the 'Create SQL Server' wizard. The first step, 'Logical server details', is displayed. The 'Server name' field contains 'server-maersk-sea-tp035262'. The 'Subscription' dropdown is set to 'Azure for Students'. The 'Resource group' section shows 'DB_Maersk_SEA_TP035362' selected under 'Use existing'. The 'Location' is set to 'Southeast Asia'. The 'Allow azure services to access server' checkbox is checked. At the bottom, there are 'Create' and 'Automation options' buttons.

Figure 23: Create SQL Server for Australia-East



The screenshot shows the 'Create SQL Server' wizard. The second step, 'Logical server details', is displayed. The 'Server name' field contains 'server-maersk-aus-tp035362'. The 'Subscription' dropdown is set to 'Azure for Students'. The 'Resource group' section shows 'DB_Maersk_AUS_TP035362' selected under 'Use existing'. The 'Location' is set to 'Australia East'. The 'Allow azure services to access server' checkbox is checked. At the bottom, there are 'Create' and 'Automation options' buttons.

Figure 24: Create SQL Server for Southeast Asia

A SQL server that used to host the database is required before the creation of SQL database. SQL servers are created in the database resource group that has established earlier for both Southeast Asia and Australia East region. The servers are placed at the similar regions according to the database resource group. Each SQL server has its own name to differentiate the server according to its hosted region. The server admin login and password are filled in for authentication purpose of the server.

4.4 Azure SQL Database

Azure SQL Database is a SQL database native to the cloud, also can be said as a platform as a service (PaaS) database which allows the complete development and deployment environment in the cloud with resources provided (Microsoft Azure, 2018). SQL database is optimized for the application development, it offers compatibility with most SQL Server features (Microsoft Azure, 2018).

In Azure SQL Database, each database is isolated from each other, each with its own service tier with a definite performance level (Microsoft Azure, 2018). It provides different performance level according to different needs of the organization (Microsoft Azure, 2018). Azure SQL database is optimized to minimize the overall expenses as it lowers the need to manage the underlying operating system and database this includes manage upgrades and backups. It is suitable for the new cloud-based applications to take advantages of the cost savings and performance optimization that managed by the cloud (Microsoft Azure, 2018). PaaS database is chosen over the IaaS database (Infrastructure as a service) because the system is new and do not have the existing on-premises application that need to extend to the cloud.

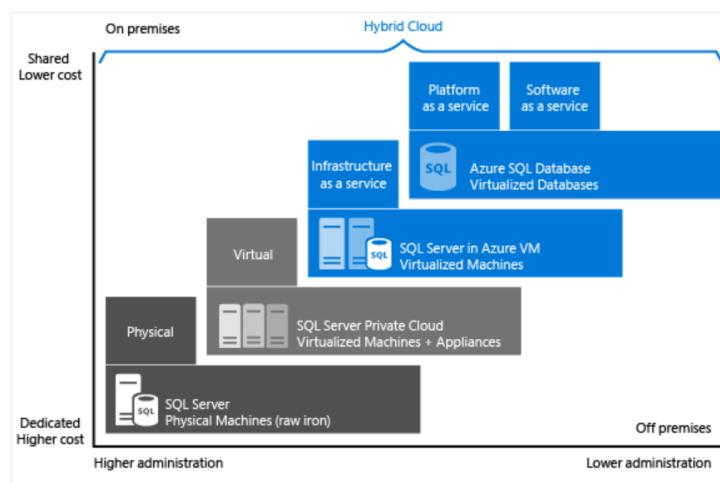


Figure 25: Characteristic of level of administration over degree of cost efficiency (Microsoft Azure, 2018)

4.4.1 Setup Primary SQL Database

After the setting up of the SQL servers, SQL database is created in the database resource group that has created earlier with the respective server that responsible to handle the SQL database for a particular region. The primary database is created with the name of “SQLDB_Maersk_SEA_TP035362” which represent as the SQL database that hosted in Southeast Asia. SQL server that hosted in Southeast Asia region is selected to linked with the database in order to successfully hosted the SQL database for the region of Southeast Asia. Blank database is selected as the source type.

The SQL database with Standard (S1) pricing tier is selected for the primary SQL database that includes with 20 DTUs (Database Transaction Units) and 250 GB storage. With Standard (S1) tier, it can drive 20 users simultaneously per second for transaction activity and able to store user data with average amount of database storage. Assumption is being made that the number of admins and agents in Maersk Line company located in Southeast Asia region is less than 20. Therefore, the Standard (S1) tier is sufficient to support the daily operation of database. The cost that required for the Standard (S1) tier that support 20 DTU is RM 125.99 per month as shown in figure above.

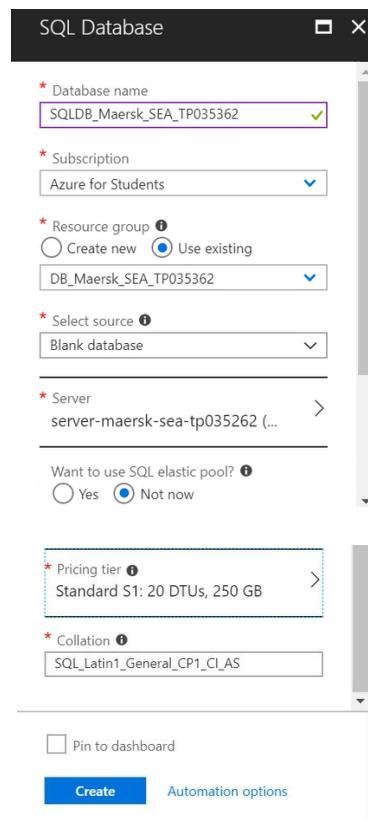


Figure 26: Create SQL Database

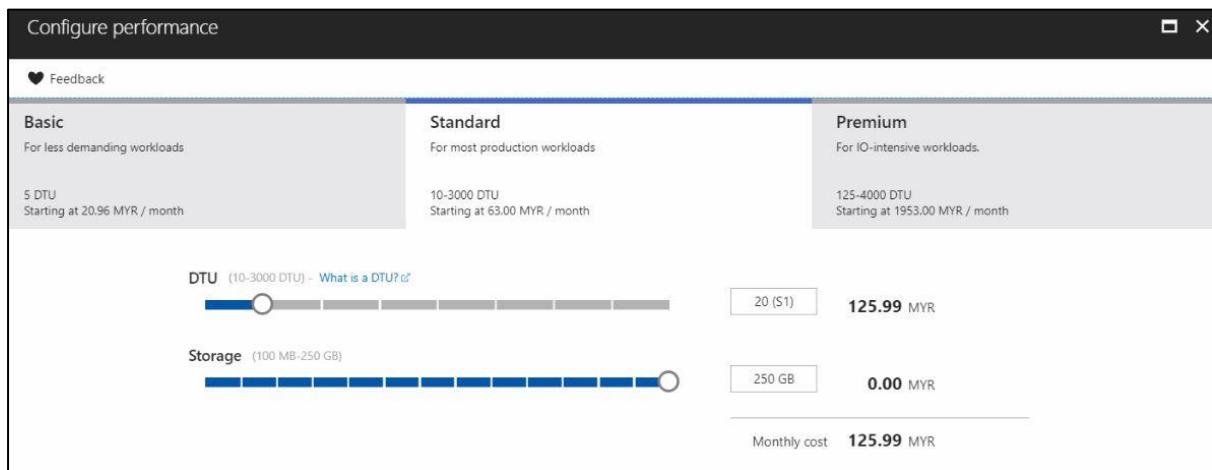


Figure 27: SQL Database Pricing Tier

4.4.2 Setup Secondary Database

The secondary database is set up in the Geo-Replication service of the primary database. Geo-Replication service produce copy of database a data centre anywhere in the world (Microsoft Azure, 2018). It used to support failover by copy the entire data in the primary database to secondary database including the day to day updates. The secondary database will be hosted in Australia-east region.

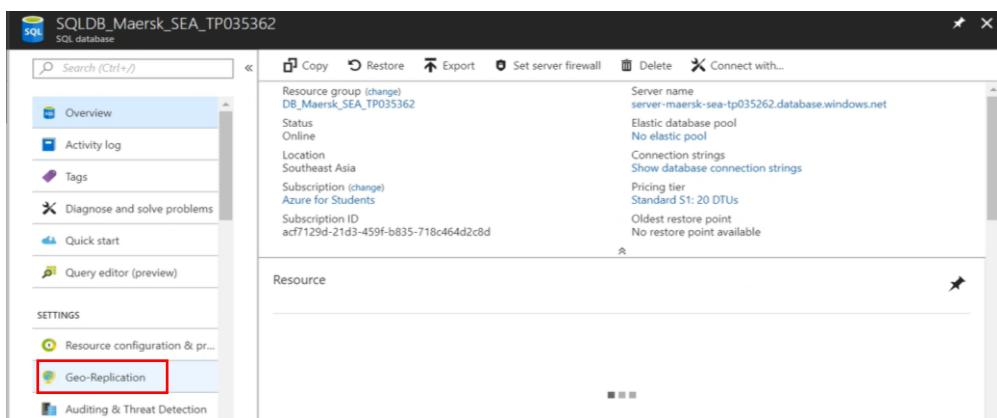


Figure 28: Geo-Replication

The secondary database is created by first selecting the target region as shown in figure below. As the secondary database will be hosted in Australia-East, “Australia East” is selected. The secondary type is fixed as readable type meaning that the data in secondary database can only be read, the data can only be update when it has swap with primary database when the primary database is failed to operate. The server is set as SQL server that used to host in Australia-East

that has created earlier. The pricing tier is selected as Standard (S1) service which is similar to the primary database.

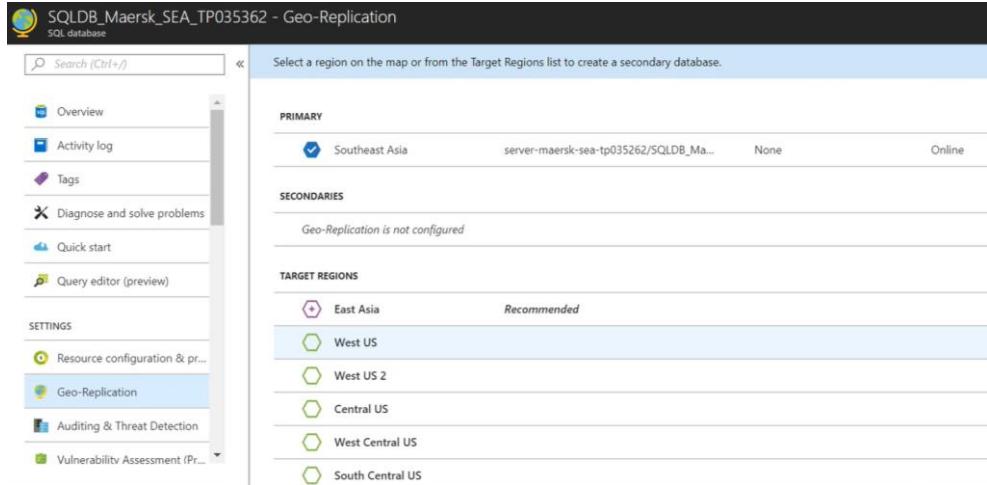


Figure 29: Select region where the secondary database is located

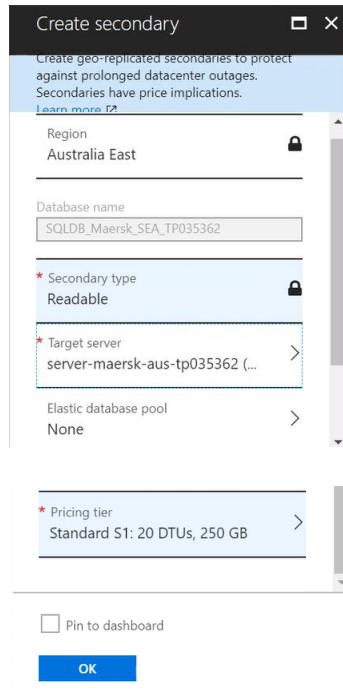


Figure 30: Create secondary database

During the process of replication and seeding process of database between primary and secondary database, the dotted line is shown on the map from Southeast Asia to Australia-East

as shown in the figure below. When, the replication and seeding process is completed the dotted line will change to solid straight line.



Figure 31: Before database replication and seeding process



Figure 32: After database replication and seeding process

4.5 Configure Failover Policy

After the completion of database replication and seeding process, failover policy is established by firstly go to the SQL server of the primary database which host in Southeast Asia and press the “Failover groups” as shown as figure below. Click on “Add group” to fill in the required criteria for failover policy configuration. The failover group with “failover-maersk-tp035362” group name is created. The secondary server which is the Australia-East server is filled in. The read and write failover policy is set as automatic with 1 hour of read and write grace period. Failover policy is created to handle the failure of primary SQL server by automatically recover the data from the secondary database to let the secondary database to temporarily replace the primary database until the primary database has recovered. This used to prevent the loss of SQL database data and maintain its service availability to the target user.

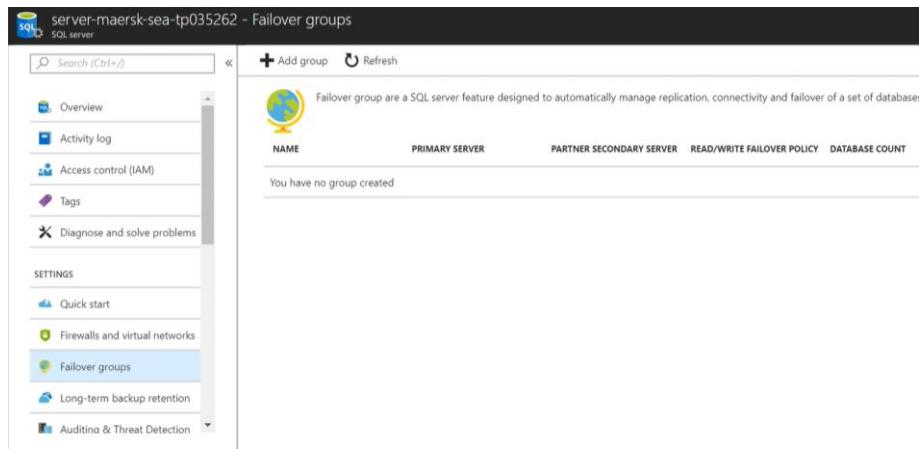


Figure 33: Failover Group

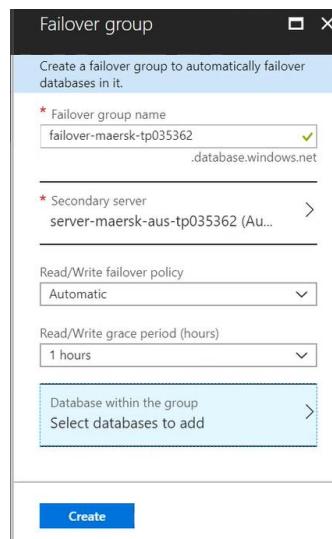


Figure 34: Create Failover Group

The SQL database is added with primary database which hosted in Southeast Asia as shown in figure below, only the primary database is added as the primary database is linked with secondary database, so the failover policy can be apply for both by just adding the primary database.

NAME	ROLE	SECONDARY SERVER	STATUS
SQLDB_Maersk_SEA_TP035362	Primary	server-maersk-aus-tp035362	Online

Selected/Eligible databases
1/1

Select

Figure 35: Select primary database for failover group configuration

The added database results will be shown in the row of “Database within the group” by altering the original display which is 0/1 to 1/1.

Create a failover group to automatically failover databases in it.

* Failover group name
failover-maersk-tp035362

* Secondary server
server-maersk-aus-tp035362 (Au...)

Read/Write failover policy
Automatic

Read/Write grace period (hours)
1 hours

Database within the group
1 / 1

Create

Figure 36: Database within the group in Failover Group

The Geo-Replication will be shown as figure below after configuring the failover policy.

	SERVER/DATABASE	FAILOVER POLICY	STATUS
PRIMARY			
<input checked="" type="checkbox"/>	Southeast Asia	server-maersk-sea-tp035262/SQLDB_Ma...	failover-maersk-tp035362 (Aut... Online
SECONDARIES			
<input checked="" type="checkbox"/>	Australia East	server-maersk-aus-tp035362/SQLDB_Maersk_SEA_TP035362	Readable ...

Figure 37: Result of successful configuring the failover policy

4.6 Firewalls and Virtual Networks

After completing the configuration of failover policy, firewalls and virtual network is established to secure the storage account to a specific set of allowed networks (Microsoft Azure, 2018). The firewall and virtual network rules is configured into each SQL server created earlier that hosted at different region to allow access of the developer in order to publish the updated version of system to the cloud platform if changes are necessary to be done. The firewalls and virtual network is done by clicking on “Add client IP” to add the client IP information to provide access to the database in the server created earlier and save it by clicking on “Save” to save the respective information.

The screenshot shows the Microsoft Azure portal interface for managing SQL servers. The title bar says "SQL servers" and "clingtonoutlook (Default Directory)". Below the title bar are buttons for "+ Add", "Edit columns", "Refresh", and "Assign Tags". A search bar and filter dropdowns for "All subscriptions", "All resource groups", "All locations", and "No grouping" are present. The main area displays a table with two items:

NAME	STATUS	LOCATION	SUBSCRIPTION
server-maersk-aus-tp035362	Available	Australia East	Azure for Students
server-maersk-sea-tp035262	Available	Southeast Asia	Azure for Students

Figure 38: SQL servers

The screenshot shows the Microsoft Azure portal for a specific SQL server named "server-maersk-sea-tp035262". The title bar says "server-maersk-sea-tp035262 - Firewalls and virtual networks". On the left, there is a sidebar with navigation links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Quick start, Firewalls and virtual networks (which is selected and highlighted in blue), Failover groups, and Long-term backup retention. The main content area has a "Save" and "Discard" button at the top right, along with a "+ Add client IP" button. It contains two informational sections:

- "Connections from the IPs specified below provides access to all the databases in server-maersk-sea-tp035262." Below this is a switch labeled "Allow access to Azure services" with options "ON" and "OFF" (set to ON). It also shows "Client IP address" as 14.192.212.232 and a table for "Firewall rules" with columns "RULE NAME", "START IP", and "END IP". A message states "No firewall rules configured."
- "Connections from the VNET/Subnet specified below provides access to all databases in server-maersk-sea-tp035262." This section is currently empty.

Figure 39: Firewalls and virtual networks

The screenshot shows the Azure portal interface for managing firewalls and virtual networks for a specific SQL server. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Quick start, Firewalls and virtual networks (which is selected and highlighted in blue), Failover groups, and Long-term backup retention. The main content area displays a summary message about allowing access to Azure services, with an 'ON' toggle switch currently set to 'ON'. Below this, a 'Client IP address' field is set to 14.192.212.232. A table lists a single IP rule: Rule Name 'ClientIPAddress_2018-4-8_1', Start IP '14.192.212.232', and End IP '14.192.212.232'. A note below the table states that connections from the specified VNET/Subnet provide access to all databases in the server.

Figure 40: Add client IP in firewalls and virtual networks

4.7 Azure Web Application Services

After setting up the firewalls and virtual networks, the Azure Web Application Services is created by going to the “App Service” session and click on the add button to add the web application service. Web Application or Web App is the service used to host the Maersk CMS web application on Azure cloud platform. As the Maersk CMS system is created using ASP.NET MVC framework with code first approach, the Web APP + SQL service is selected to support the code first approach and ensure that the success of the Maersk CMS system to be published to cloud.

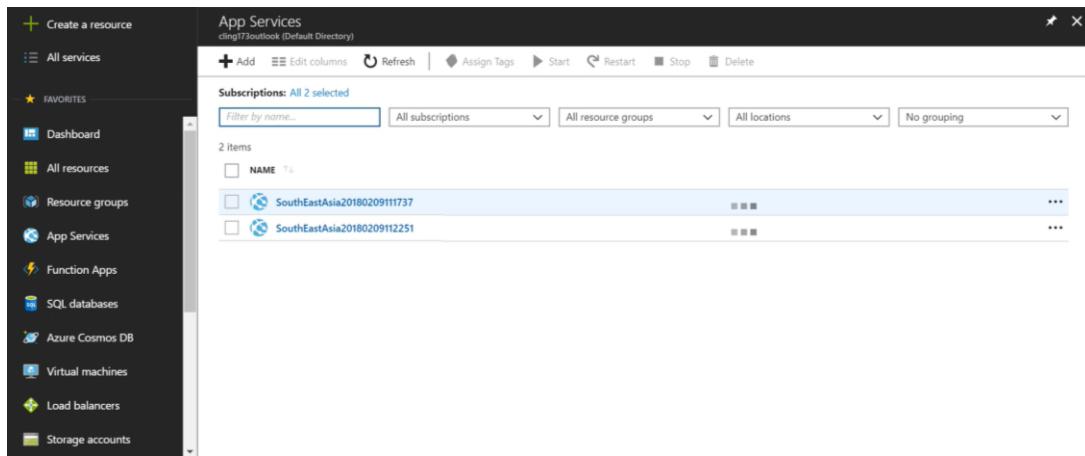


Figure 41: App Services

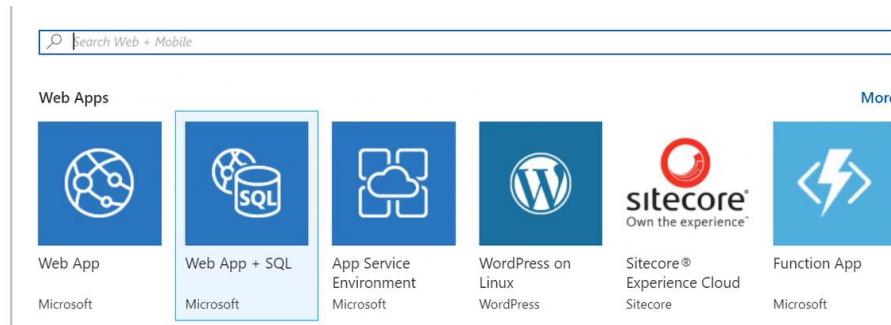


Figure 42: Web App + SQL apps service

After that, the required information need to be filled in as shown in figure below. The App name named as “Maersk-SEA-TP035362” is created for the web application that will be hosted in Southeast Asia region. The App name will later be set as the URL of the web application after the system is published. The resource group created earlier that used to hold the web application service is used where the resource group with representation of corresponding regions are selected.

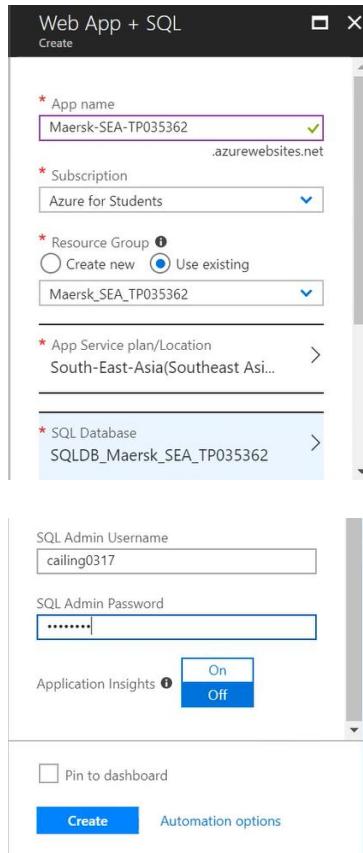


Figure 43: Create Web App + SQL

A new App Service Plan is created according to the needs of the business by clicking on the arrow at the “App Service Plan/ Location” column. The App Service plan name is being created according to the location specify earlier. The pricing tier of S1 (standard) is being selected for both Southeast Asia and Australia East regions. This pricing tier is selected because it able to provide sufficient support to meet the business requirements of Maersk Line CMS system with moderate operation this includes with specification such as Single core, 1.75 GB RAM, 50 GB storage, daily back up and so forth. It also the minimum plans that supports the traffic manager services that can direct user from different region to the nearest web app server to be used which improved the performance of the application.

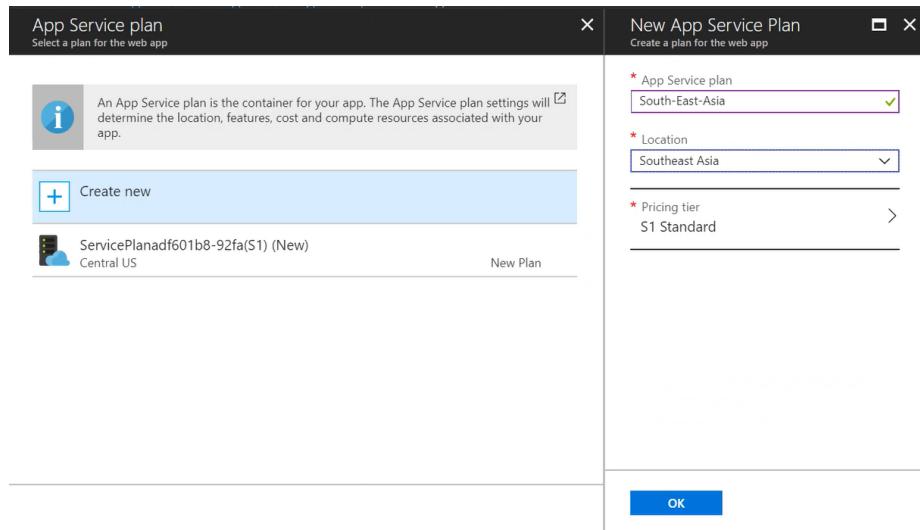


Figure 44: Create App service plan

Choose your pricing tier Browse the available plans and their features		
MYR/MONTH (ESTIMATED)	MYR/MONTH (ESTIMATED)	MYR/MONTH (ESTIMATED)
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
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
Select		

Figure 45: Type of services plans

Then, the SQL Database is selected from the SQL database created earlier, the SQL Admin Username and Password is filled in with the Admin username and password that has registered earlier for SQL authentication purpose. The same steps that has discussed above are repeated for the creating of web app for Australia-East region with different App name which will be named as “Maersk-AUS-TP035362”.

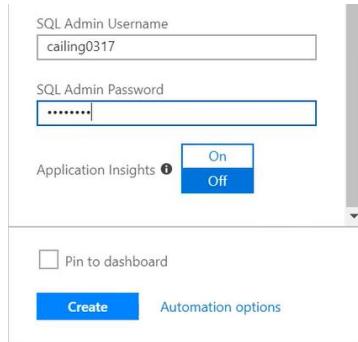


Figure 46: SQL Admin Username and Password

The created App Service and App Service Plan is shown as figure below in each of the web application resource group create earlier with specified region.

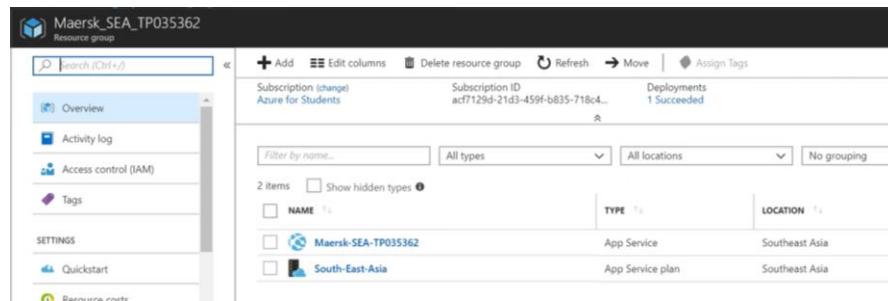


Figure 47: Result shown after successfully create the app services

After that, go to the app service that has created just now and clicked on it. After that, click on the application setting to do some modification towards the connection strings. In the default connect of the connection string, the default user ID is changed to the registered user ID as done earlier during the setup of SQL server to enable the connection of SQL database of the web application can be well connected with the Azure SQL database. The steps are done for the app service created for both of the Southeast Asia and Australia-East regions.

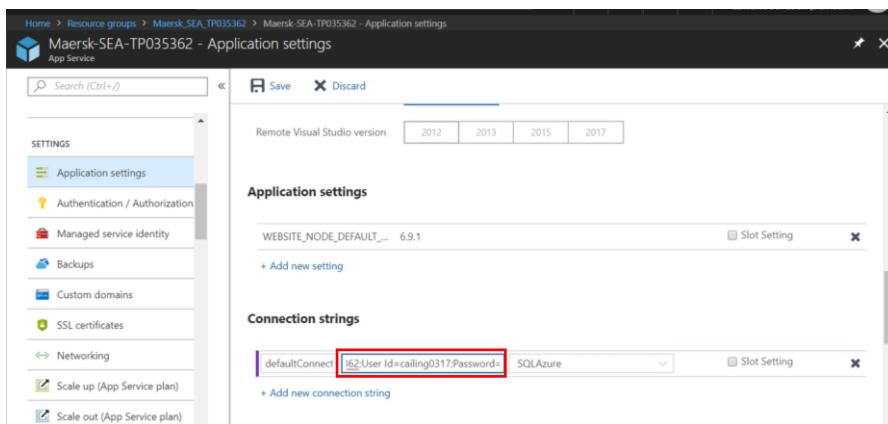


Figure 48: Change Connection strings

4.8 Deploy ASP.NET Web Application

After the completion of setting up the web app service, SQL server and database, the Maersk Line CMS system which is a ASP.NET web application is ready to publish to Azure cloud platform. Firstly, click on the solution explorer, right click on the project developed and select “Publish” as shown in figure below.

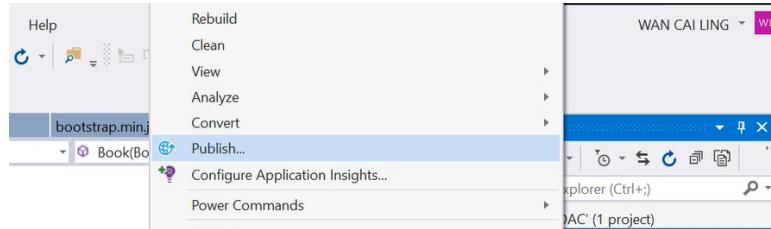


Figure 49: Publish ASP.NET web application

Then, select “Select Existing” and then click on the arrow at the publish button and select create profile to allow the execute of code first migration selection during setting which will be done in the following steps.

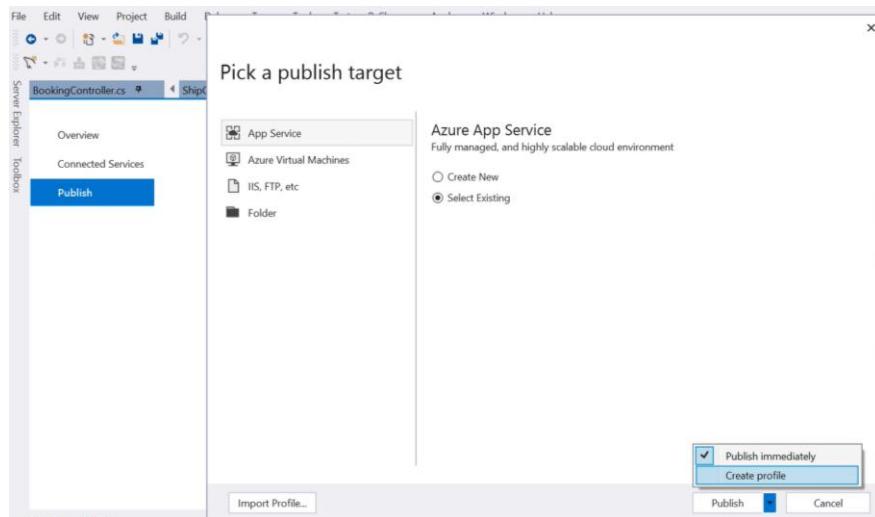


Figure 50: Select publish target

After that, choose the web app service (Maersk- SEA-TP035362) which represent the web application that need to publish in Southeast Asia region in the resource group created in Azure portal earlier.

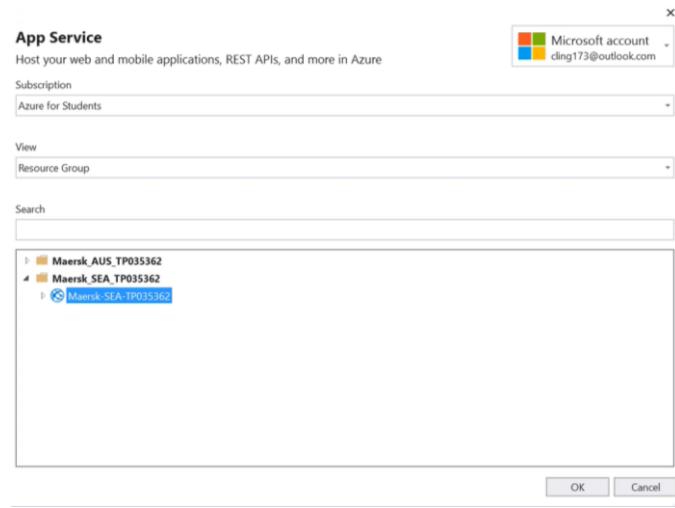


Figure 51: Select App Service to be published

Then click on setting to set up the publish setting included with connection and settings session. In connection session, connection settings are auto configured by Visual Studio as shown in figure below, the connection can be validated by clicking on the “Validate Connection” button.

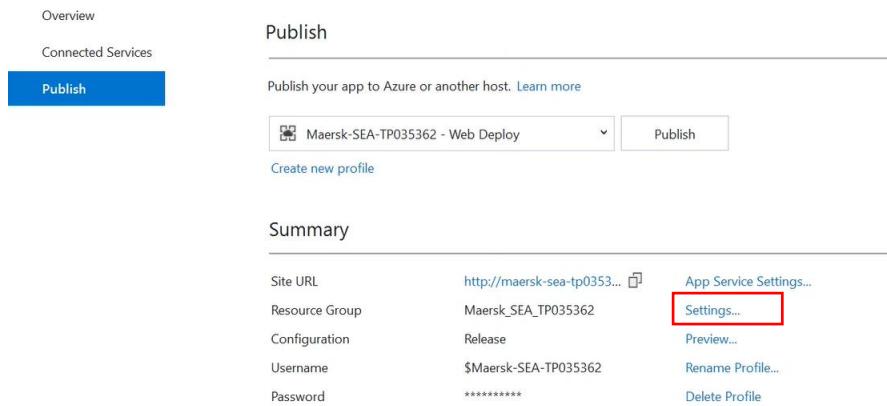


Figure 52: Publish settings

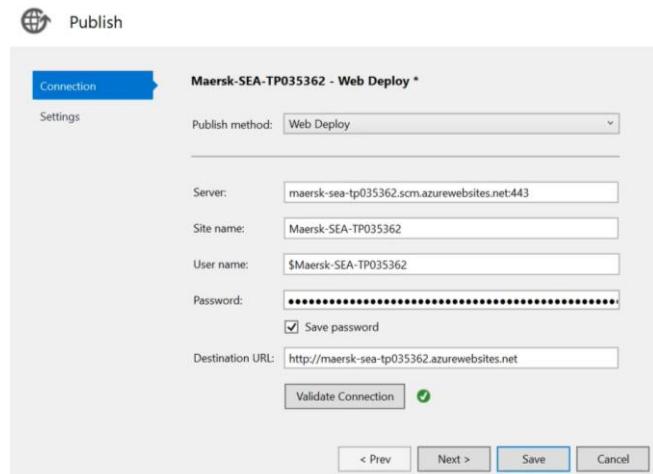


Figure 53: Publish connection configuration

After the connection has validated, go to the setting session and set the configuration to “Release” and tick on the “Execute Code First Migrations” to allow the web application to run all the migration when it is first started as the Maersk Line CMS system database is developed using code first approach where the database application are transform from coded classes. Through this, Visual Studio will automatically produce database schema and create database to Azure SQL database. The Application Db Context (Default Connection) is automatically configure by Visual Studio and the details can be check by clicking on the button beside. Click on “Save” button after all the step has been done and the information has been checked.

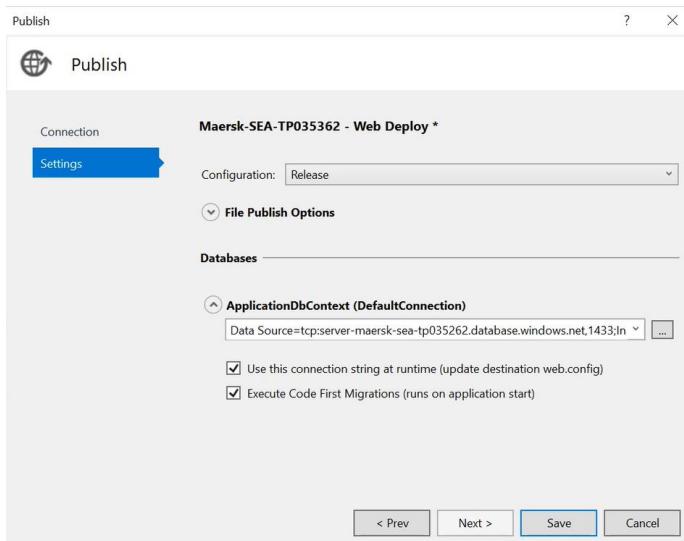


Figure 54: Publish settings configuration

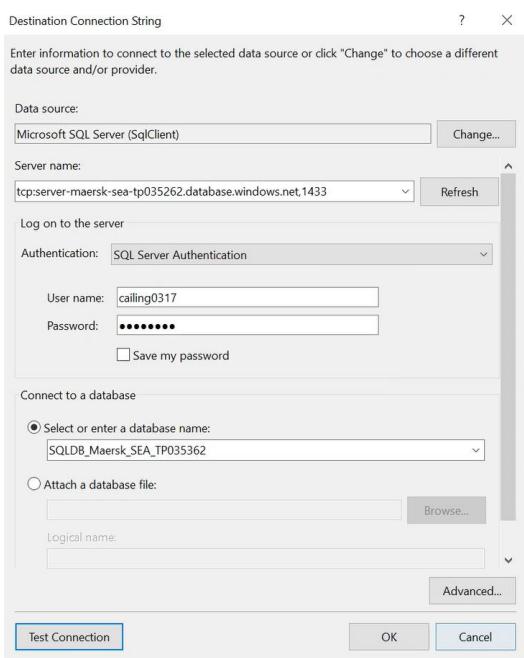


Figure 55: Test connection

Click on publish button to deploy the web application to Azure cloud platform.

The screenshot shows the 'Publish' blade in the Azure portal. At the top left are navigation links: 'Overview', 'Connected Services', and a blue 'Publish' button. The main title is 'Publish'. Below it is a sub-header: 'Publish your app to Azure or another host. [Learn more](#)'. A dropdown menu shows 'Maersk-SEA-TP035362 - Web Deploy' with a downward arrow, and a red box highlights the 'Publish' button to its right. Below the dropdown is a link 'Create new profile'. The section title 'Summary' is followed by a table of deployment details:

Site URL	http://maersk-sea-tp035362	App Service Settings...
Resource Group	Maersk_SEA_TP035362	Settings...
Configuration	Release	Preview...
Username	\$Maersk-SEA-TP035362	Rename Profile...
Password	*****	Delete Profile

Figure 56: Publish web application to Azure Cloud platform

The steps are repeated for the web application which will be hosted in Australia East region, by publishing its respective web app. After successful deploy on both of the web applications, the Maersk Line CMS system can be access by accessing the web app URL which are “maersk-sea-tp035362.azurewebsites.net” and “maersk-aus-tp035362.azurewebsites.net”.

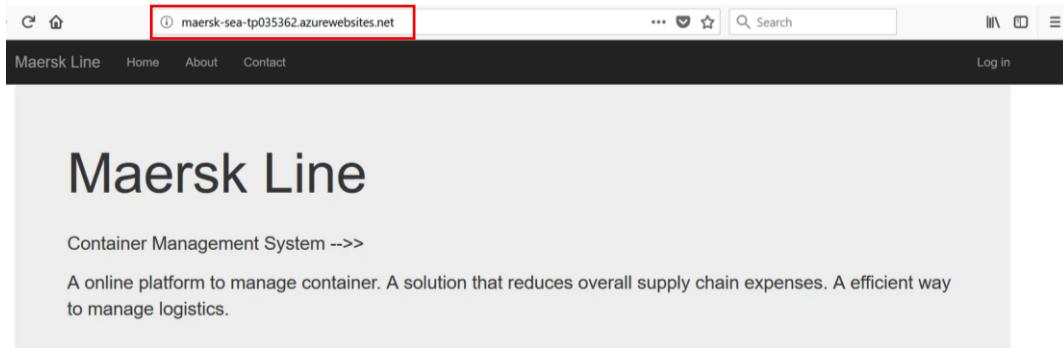


Figure 57: Web URL for web application in Southeast Asia region

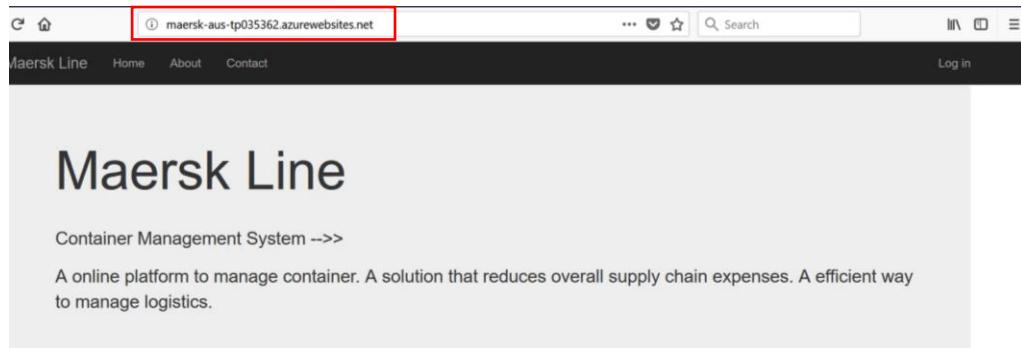


Figure 58: Web URL for web application in Australia-East region

4.9 Azure Traffic Manager

Traffic Manager allows the control of the distribution of user traffic for service endpoints in different data centres. Traffic manager monitor the endpoint by directing the client request to the most suitable end point which is the nearest web app service to avoid web congestion. Besides, traffic manager also supports automatic failover management through the detect of the failed endpoint and redirect to another nearest well-performed endpoint to maintain the web application availability.

4.9.1 Setup Traffic Manager Profile

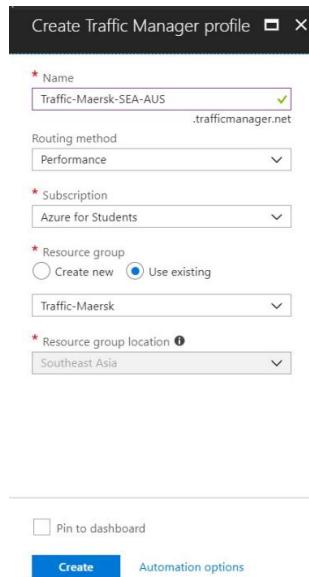


Figure 59: Create Traffic Manager profile

Firstly, go to “Traffic Manager Profile” and click on “Add”. The name for traffic manager is set as “Traffic-Maersk-SEA-AUS” which represent that this traffic manager is responsible for Southeast Asia and Australia-East regions. The URL is created from the name which is “Traffic-Maersk-SEA-AUS.trafficmanager.net” in order to connect to the client. Routing method is set to “Performance” which the web traffic will be managed based on performance by direct the traffic to the web application that is closest to the user based on the user current location. The performance measurement will be based on the measurement of network latency between user and application (endpoint). This improves the performance of the web application by directing traffic to the endpoint with the lowest network latency for the client automatically. The resource group created earlier for traffic manager is selected to hold the content regarding the traffic manager.

4.9.2 Setup Endpoint

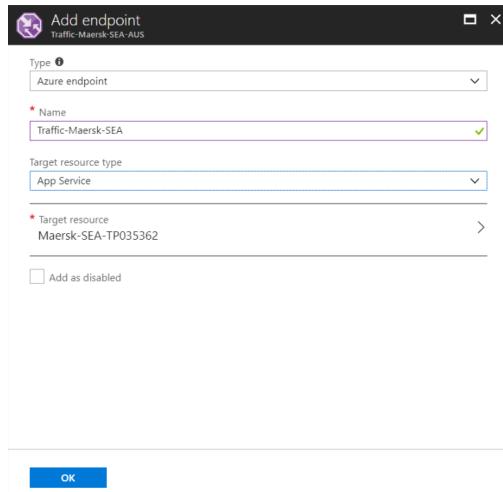


Figure 60: Add endpoint for Southeast Asia

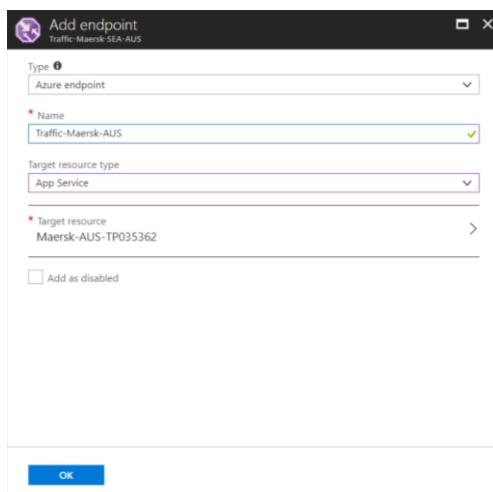


Figure 61: Add endpoint for Australia-East

The URLs of the Web Apps created earlier which are “maersk-sea-tp035362.azurewebsites.net” and “maersk-aus-tp035362.azurewebsites.net” act as the endpoints required for the traffic manager. The setup is done in “Add Endpoint”, the type is set to “Azure Endpoint”, the name of endpoints is named according to the specified region of the web app. The target resource type is App Service create earlier and select according to the specified region. The web Apps Services created earlier is linked with the target resource.

5.0 Testing

5.1 Unit Testing

5.1.1 Login

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
LO-01	Admin and agent login with correct email and password.	Email: valid_Email@outlook.com Password: validPassword@123	Login successful. Direct to home page with either admin or agent tasks according to the logged user account.	As Expected
LO-02	Admin and agent login with wrong email.	Email: invalid_Email@outlook.com Password: validPassword@123	Login unsuccessful. Display Error Message.	As Expected
LO-03	Admin and agent login with wrong password.	Email: valid_Email@gmail.com Password: invalidPassword@123	Login unsuccessful. Display Error Message.	As Expected
LO-04	Admin and agent login with empty input.	Email: Empty Input Password: invalidPassword@123	Login unsuccessful. Display Error Message.	As Expected

Table 13: Login Test Plan

5.1.2 Booking Ship Schedule

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
BO-01	Displayed the selected book schedule information selected by agent and the booking criteria to perform booking task.	-	The selected book schedule information is displayed along with the required booking criteria that need to be filled in.	As Expected
BO-02	Input all the required booking criteria with valid information.	Customer: Lily Lim Container Type: Liquid Bay used: 20	Booking successful. Display booking successful message.	As Expected
BO-03	Input the required information with invalid information.	Customer: Pick Customer Container Type: 123 Bay used: abc	Booking unsuccessful. Display Error Message.	As Expected
BO-04	Empty input for the booking required criteria.	Customer: Empty Input Container Type: Liquid Bay used: Empty Input	Booking unsuccessful. Display Error Message.	As Expected
BO-05	The cargo size is insufficient to sustain the bay to be used.	Bay used: 120 Remaining cargo size: 100	Booking unsuccessful. Display Error Message.	As Expected

Table 14: Booking Ship Schedule Test Plan

5.1.3 View Customer

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
VC-01	Click on view customer.	-	Display the overview of customer information.	As Expected

Table 15: View Customer Test Plan

5.1.4 Register Customer

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
RE-01	Input all the required register criteria with valid information.	Customer Name: John Lim IC Number: 921112193322 Contact Number: 01199992222 Email: john@hotmail.com	Register customer successful. Direct to customer index page and displayed the list of all the customer information together with the added customer information.	As Expected
RE-02	Input the required register criteria with invalid information.	Customer Name: 123 IC Number: abc Contact Number: 011999aacc Email: hotmail.com	Register customer unsuccessful. Display Error Message.	As Expected

RE-03	Do not complete filled in all the required information by leaving empty input.	Customer Name: Empty Input IC Number: 921112193322 Contact Number: Empty Input Email: john@hotmail.com	Register customer unsuccessful. Display Error Message.	As Expected
-------	--	---	---	-------------

Table 16: Register Customer Test Plan

5.1.5 View Ship

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
VS-01	Click on view ship.	-	Display the overview of ship information.	As Expected

Table 17: View Ship Test Plan

5.1.6 Add Ship

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
AS-01	Input all the required add ship criteria with valid information.	Ship Name: Golden Ship Ship Starting Point: Australia Ship Destination: Malaysia Cargo Size: 300	Add Ship successful. Direct to ship index page and displayed the list of all the ship information together with the added ship information.	As Expected

AS-02	Input the required register criteria with invalid information.	Ship Name: 123 Ship Starting Point: 456 Ship Destination: 789 Cargo Size: aaa	Add ship unsuccessful. Display Error Message.	As Expected
AS-03	Do not complete filled in all the required information by leaving empty input.	Ship Name: Empty Input Ship Starting Point: Australia Ship Destination: Malaysia Cargo Size: Empty Input	Add ship unsuccessful. Display Error Message.	As Expected

Table 18: Add Ship Test Plan

5.1.7 View Schedule

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
VE-01	Click on view schedule.	-	Display the overview of schedule information.	As Expected

Table 19: View Schedule Test Plan

5.1.8 Add Schedule

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
AE-01	Input all the required add schedule criteria with valid information.	Origin: Taipei Destination: Malaysia Ship Departure Date: 04/18/2018 10:35 AM Ship Arrival Date: 04/25/2018 10:40AM	Add Schedule successful. Direct to schedule index page and displayed the list of all the ship schedule information together with the added schedule information.	As Expected
AE-02	Do not complete filled in all the required information by leaving empty input.	Origin: Empty Input Destination: Malaysia Ship Departure Date: 04/18/2018 10:35 AM Ship Arrival Date: 04/25/2018 10:40AM	Add ship unsuccessful. Display Error Message.	As Expected

Table 20: Add Schedule Test Plan

5.1.9 Edit Schedule

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
EC-01	Edit the ship schedule with valid information.	Origin: Taipei Destination: Malaysia Ship Departure Date: 04/17/2018 08:00 PM Ship Arrival Date: 04/25/2018 10:40AM	Edit schedule successful. Direct to schedule index page and displayed the list of all the ship schedule information together with the edited schedule information.	As Expected
EC-02	Empty the required input.	Origin: Taipei Destination: Malaysia Ship Departure Date: Empty Input Ship Arrival Date: 04/25/2018 10:40AM	Edit schedule unsuccessful. Display Error Message.	As Expected

Table 21: Edit Schedule Test Plan

5.1.10 View Booking

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
VB-01	Click on view booking.	-	Display the overview of all booking information.	As Expected

Table 22: View Booking Test Plan

5.1.11 Register Agent

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
RA-01	Input all the required register criteria with valid information.	Email: yenyen@outlook.com Password: Yenyen@123 Confirmed Password: Yenyen@123 Agent Name: Yen Yen Company Name: BTS Group	Register agent successful. Direct to admin home page.	As Expected
RA-02	Input the required register criteria with invalid information.	Email: yenyen.com Password: 123 Confirmed Password: 123 Agent Name: Yen Yen Company Name: BTS Group	Register agent unsuccessful. Display Error Message.	As Expected
RA-03	The input of password and confirm password varies.	Password: YenYen@123 Confirmed Password: Yen123	Register agent unsuccessful. Display Error Message.	As Expected
RA-04	Do not complete filled in all the required	Email: Empty Input Password: Yenyen@123	Register agent unsuccessful.	As Expected

	information by leaving empty input.	Confirmed Password: Yenyen@123 Agent Name: Yen Yen Company Name: BTS Group	Display Error Message.	
--	-------------------------------------	--	------------------------	--

Table 23: Register Agent Test Plan

5.1.12 Log Off

Test Case ID	Test Case	Test Data	Expected Result	Actual Result
LO-01	Click on log off.	-	Return to main page.	As Expected

Table 24: Log Off Test Plan

5.2 Performance Testing

Microsoft Azure support performance testing feature which allows user to check the web application performance and ensure that the application can handle the traffic during peak usage. The performance testing is done in a non-production environment by having the web application use an App Service plan, through this, the existing customer will not be affected or slow down the web application in production (Microsoft Azure, 2018). The feature can be found in Web App service under “Performance Test” and click on “New to start a new performance test. The test type is set as “Manual Test”, the traffic manager URL is added automatically by Azure for testing purpose. A test name is given which is “MaerskPerfTest02”, the generated load will be come from the Southeast Asia where the target users are located. The testing is performed for 250 concurrent users within 5 minutes.

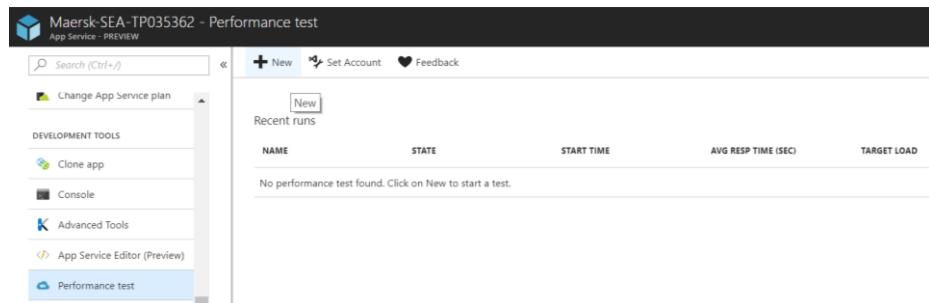


Figure 62: Performance Testing

The image shows two overlapping dialog boxes. The left dialog is titled "New performance test" and has a "PREVIEW" badge. It contains fields for "NAME" (set to "MaerskPerfTest02"), "GENERATE LOAD FROM" (set to "Southeast Asia (Web app Location)", "USER LOAD" (set to "250"), and "DURATION (MINUTES)" (set to "5"). At the bottom are "Run test" and "Done" buttons. The right dialog is titled "Configure test using" and also has a "PREVIEW" badge. It contains a "TEST TYPE" dropdown set to "Manual Test" and a "URL" input field containing "http://traffic-maersk-sea-aus.trafficmanager....".

Figure 63: Fill in all criteria performance testing

The performance test need to take about 15 minutes to complete, the test status will change to “Completed” when the testing is finished.

The screenshot shows the Azure App Service Performance test preview interface. At the top, it displays the application name "Maersk-SEA-TP035362 - Performance test" and the status "App Service - PREVIEW". Below this is a search bar and navigation links for "Overview", "Activity log", "Access control (IAM)", and "Tags". The main area is titled "Recent runs" and lists a single entry: "MaerskPerfTest02" with a status of "Completed" (indicated by a green checkmark), started on "4/12/2018, 12:05 AM", an average response time of "0.14", and a target load of "250".

Figure 64: Performance testing complete state

The testing result generated will be shown as figure below. The result includes a pie chart representing the successful and failed requests. From the result shows in pie chart, there are total of 67500 requests were sent and 100% of them were successful to be executed.

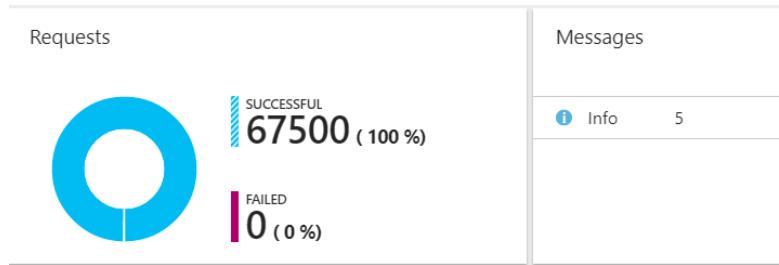


Figure 65: Success and fail requests

In the performance under load graph, shown in figure below the result shows the average response time of the web application is 0.14 seconds, running 250 user loads and serving 225 requests per second measured from time to time. The metrics at specific time can be viewed by just hovering the mouse to any point shown in the graph. From the result, the page load of the web application is slow at first and becoming faster afterwards. The respective solution should be generated to ensure the average response time of the web application can be fast and stable from the beginning.

Performance under load

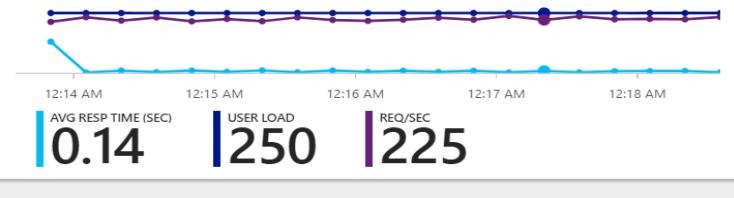


Figure 66: Performance under load result

The performance measures also include with CPU time and memory working set past hour as shown in figure below. The results show that in an average of 168 CPU time the memory working set is 4.56 B. The further details of the results can be view by clicking on the and the details will be shown in the right side. The metrics at specific time can be viewed by just hovering the mouse to any point shown in the graph. This is used to examine the CPU performance over time.

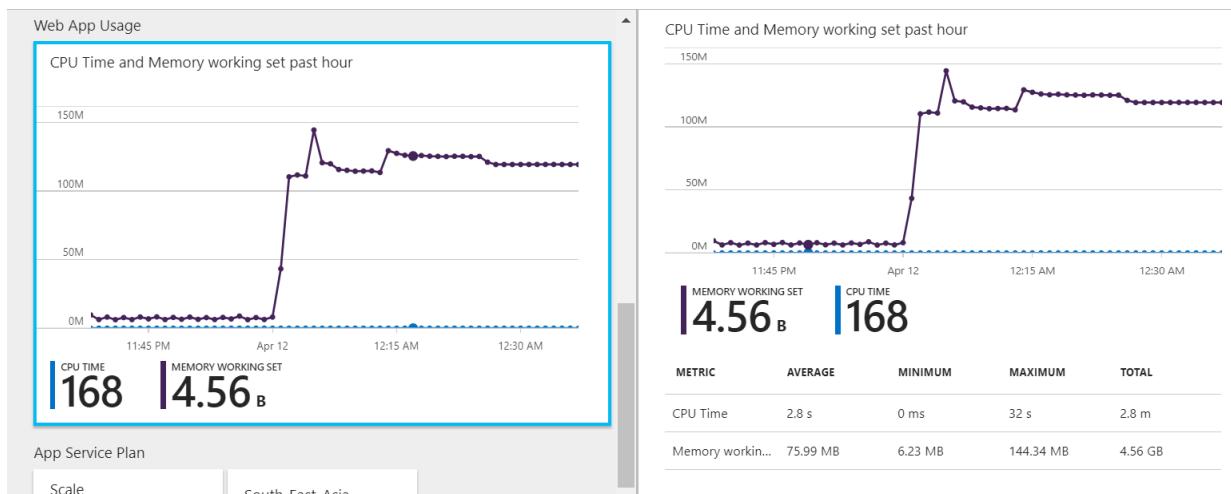


Figure 67: CPU time and memory working set past hour result

6.0 Conclusion

As conclusion, the developer has successfully completed with the Maersk Line Container system design, development and finally deployed to the Microsoft Azure cloud platform. From this project, the developer has gains knowledge towards the system deployment in Microsoft Azure cloud platform aside from the software design and development. The web application deployed to cloud platform is an effective solution which required less expenses as the resources are provided by cloud according to service type. The web application is deployed and connected to the SQL server and database that hosted in Azure, hence, the company do not need to have their own SQL server and database hardware component in order to expand the business. Besides, the web application service is hosted in different regions such as Southeast Asia and Australia-East region and connected together with endpoints to ensure system high availability and performance that able to provide user with great experience and at the same time the shipping operation can be managed efficiently. Aside from that, Microsoft Azure cloud platform support failover management through geo-replication and failover group services which prevent any data loss from the database, providing user with high availability of database services.

After finished deployed, the developer has conducted unit testing and performance testing to ensure the quality of the web application. Performance test is the function provided in Azure, troubleshooting of problem can be done easily and quickly by viewing on the statistics generated from the testing results. From the testing results shown, decision making can be done more accurately and quickly that may greatly help to increase the development of business. At last. the Maersk Line Container Management System (CMS) is consider as successfully meets its aim through utilizing cloud platform which may reduce the overall supply chain expenses and at the same time the logistics is well-managed efficiently.

7.0 References

- Kayal, S. (2014). Code First Approach in Entity Framework using Data Annotation and Fluent API. [online] Code Project. Available at: <https://www.codeproject.com/Tips/793159/Code-First-Approach-in-Entity-Framework-using-Data> [Accessed 3 Apr. 2018].
- Microsoft Azure. (2018). Choose a cloud SQL Server option: Azure SQL (PaaS) Database or SQL Server on Azure VMs (IaaS). [online] Available at: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-paas-vs-sql-server-iaas> [Accessed 6 Apr. 2018].
- Microsoft Azure. (2018). Configure Azure Storage Firewalls and Virtual Networks. [online] Available at: <https://docs.microsoft.com/en-us/azure/storage/common/storage-network-security> [Accessed 7 Apr. 2018].
- Microsoft Azure. (2018). Load test with the Azure portal. [online] Available at: <https://docs.microsoft.com/en-us/vsts/load-test/app-service-web-app-performance-test?view=vsts> [Accessed 8 Apr. 2018].
- Microsoft Azure. (2018). Overview: Failover groups and active geo-replication. [online] Available at: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-geo-replication-overview> [Accessed 6 Apr. 2018].
- Microsoft Azure. (2018). What is the Azure SQL Database service? [online] Available at: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-technical-overview> [Accessed 5 Apr. 2018].

8.0 Appendix

1. Source code of Maersk Line CMS (GitHub repository)

- <https://github.com/WanCaiLing/DDAC-TP035362>

2. Azure & System Demonstration Video

- The video can be accessed on:

<https://web.microsoftstream.com/video/3b20885d-50f8-48f3-95b4-29c3af9485c8>

3. Azure Web App URL

- <https://maersk-sea-tp035362.azurewebsites.net> (Southeast Asia region)
- <https://maersk-aus-tp035362.azurewebsites.net> (Australia-East region)

4. Sample User Credential for Testing

- Admin Account
 - Email: admin@admin.mail.com
 - Password: Admin@123
- Agent Account
 - Email: cling@hotmail.com
 - Password: Cling@123