

|  |  |  |
| --- | --- | --- |
| **Module Code** | : | CT071-5-3-DDAC |
| **Module Name** | : | Designing & Developing Cloud Applications |
| **Project Title** | : | Maersk Line Container Management System |
| **Student ID** | : | TP038288 |
| **Student Name** | : | Dylan Toh Voo Chieng |
| **Intake Code** | : | UC3F1706SE |
| **Lecturer Name** | : | Dr. Kalai Anand A/L Ratnam |
| **Hand in Date** | : | 13 April 2018 |

Contents

[Acknowledgement 4](#_Toc511368404)

[2 Introduction 5](#_Toc511368405)

[2.1 Project background 5](#_Toc511368406)

[2.2 Objective & Scope 6](#_Toc511368407)

[2.3 Requirement specifications 6](#_Toc511368408)

[3 Project plan 7](#_Toc511368409)

[4 Design 8](#_Toc511368410)

[4.1 Design Consideration 8](#_Toc511368411)

[4.2 Modelling 8](#_Toc511368412)

[4.2.1 Use Case Diagram 8](#_Toc511368413)

[4.2.2 Use case specification 9](#_Toc511368414)

[4.2.3 Sequence diagram 19](#_Toc511368415)

[4.2.4 Class diagram 31](#_Toc511368416)

[4.2.5 ERD Diagram 32](#_Toc511368417)

[4.3 Cloud Architectural Diagram 33](#_Toc511368418)

[5 Implementation 35](#_Toc511368419)

[5.1 Application Development 35](#_Toc511368420)

[5.2 Azure Publishing 37](#_Toc511368421)

[5.2.1 App service publish 37](#_Toc511368422)

[5.2.2 Database publish 40](#_Toc511368423)

[5.3 Application Scaling 44](#_Toc511368424)

[5.3.1 SQL Database Scaling 44](#_Toc511368425)

[5.3.2 Web application scaling 45](#_Toc511368426)

[5.4 Reliability and performance 46](#_Toc511368427)

[6 Test Plan & Testing Discussion 48](#_Toc511368428)

[6.1 Test Plan 49](#_Toc511368429)

[6.1.1 Login 49](#_Toc511368430)

[6.1.2 Register as admin 50](#_Toc511368431)

[6.1.3 Register as Agent 51](#_Toc511368432)

[6.1.4 Add New Shipping Schedule 52](#_Toc511368433)

[6.1.5 Add new vessel 53](#_Toc511368434)

[6.1.6 Manage Agents 53](#_Toc511368435)

[6.1.7 Add new customers 54](#_Toc511368436)

[6.1.8 New shipment 54](#_Toc511368437)

[6.2 Performance testing 56](#_Toc511368438)

[7 Azure Managed Database (Paas) 60](#_Toc511368439)

[8 Conclusion 63](#_Toc511368440)

[9 Reference 64](#_Toc511368441)

# Acknowledgement

Firstly, I would like to express my gratitude to DDAC lecturer, Dr. Kalai for sharing his knowledge and guiding me throughout this semester. I have learned a lot about the cloud computing, as well as setting up and hosting my application in Azure hosting server. Other than that, I would like to thank my fellow classmate who provided guidance and support that keeps me in smooth progress. We helped each other by discussing our progress and checking each other’s work. Ultimately, we have each other to perform testing and ensure that the system is working fine before hosting it using Azure.

# Introduction

## 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 centers 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.

## Objective & Scope

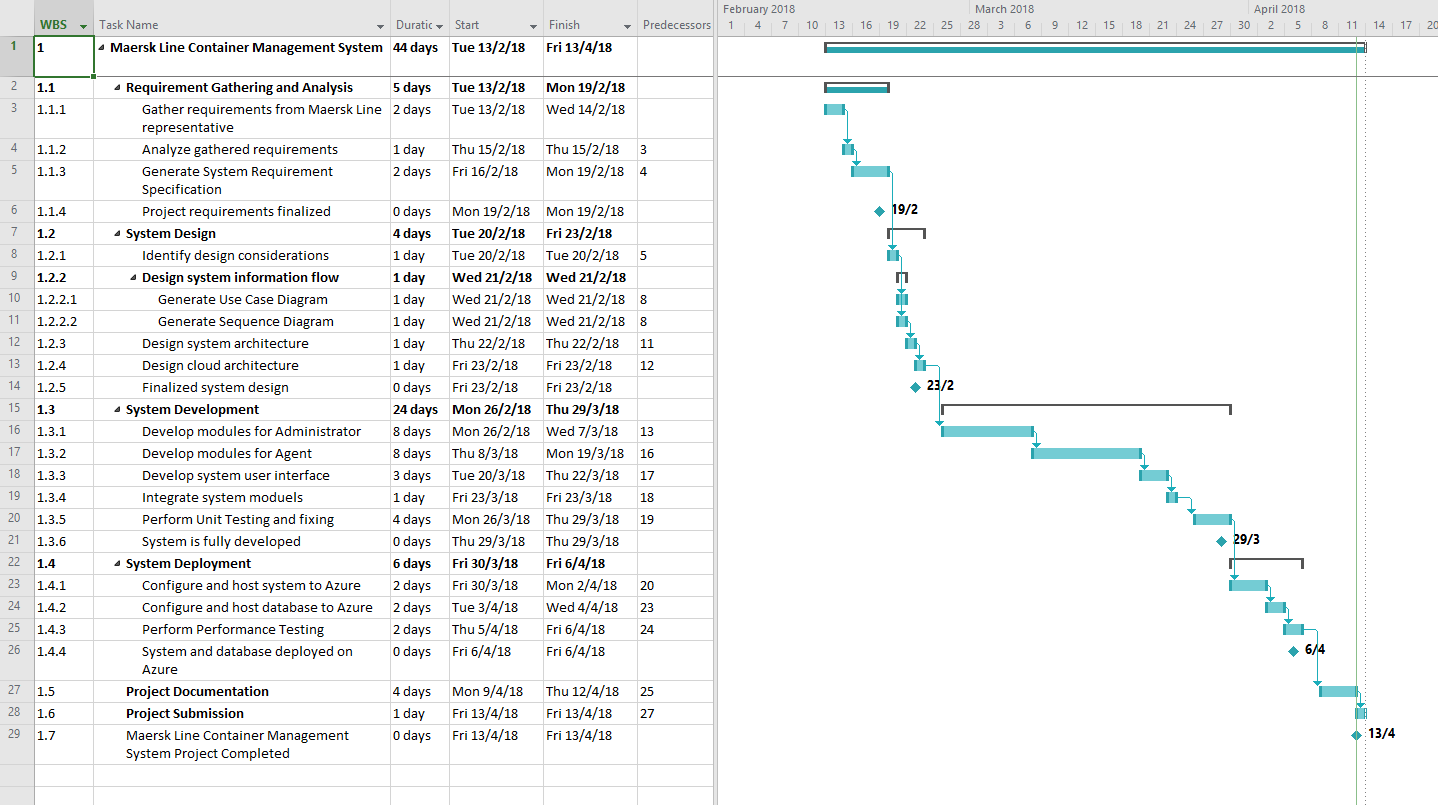
* Demonstrate the understanding of application in cloud computing in its different forms and how Microsoft Azure fits in the cloud computing space.
* Explore the Microsoft Azure development environment.
* Implement, design & deploy Maersk Line Container Management Web Application Service
* To architecturally design efficient Maersk Line Container Management Web Application Service by using Microsoft Azure as the public cloud platform.

## Requirement specifications

To design and develop a single tenant web solution for that meets the following criteria as below:

* From import, export and transhipment processing to gate operation
* To be able to scale the solution to meet the needs of demands during peak seasons
* Improve profitability, reduce costs, increase productivity, eradicates error and optimizes resources to future-proof the cargo handling business for high performance.
* Assurance & reliability through Failover Management
* Accurately allocates inbound containers to yard locations and plan outbound containers to individual haulier vehicles, delivering an exceptional level of automation and removing human error.
* Manage the entire booking process from schedule search to booking confirmation

# Project plan



# Design

## Design Consideration

There are some assumption or consideration have been made throughout the project such as listed below:

* One agent can only book one container in one shipment
* One time can only be included in one container

## Modelling

### Use Case Diagram

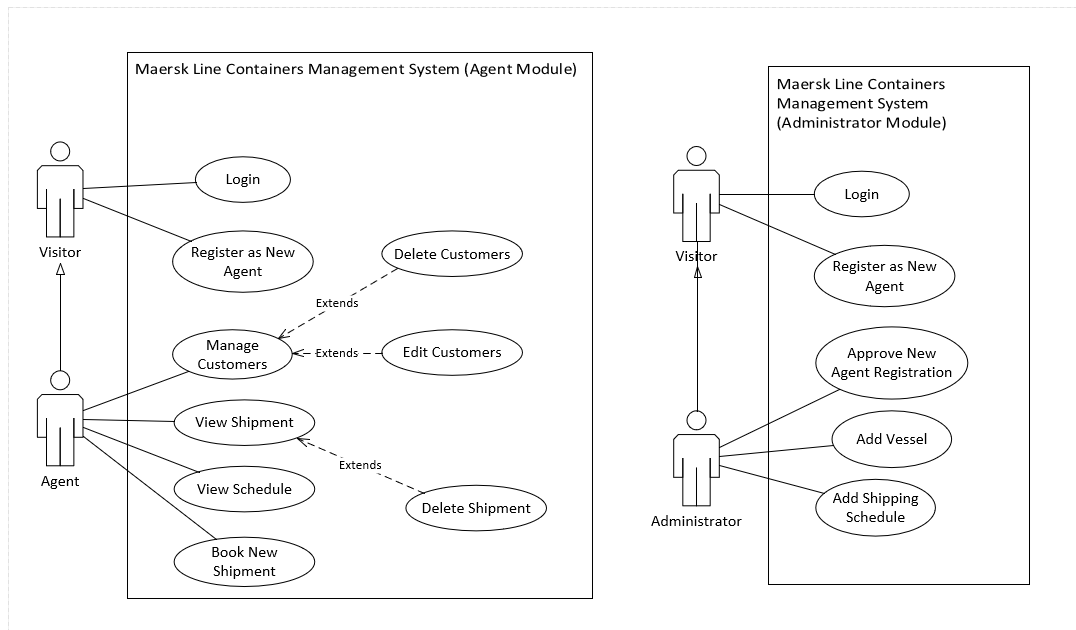


Figure 1: Use case diagram

### Use case specification

|  |  |
| --- | --- |
| Use Case: | Login |
| Summary: | An admin or agent enter their unique username and password in order to access more function of the system. |
| Dependency: | - |
| Actors: | Admin and Agent |
| Pre-condition: | Actor must be an admin or agent. |
| Main Flow: | 1. Actor enter their username and password. 2. System will search through database whether the username and password are existed and matched. 3. Actor may enter the website if they entered the correct username and password. |
| Alternative Flow: | **Step 2**: System could not be login and ask for the actor to enter the correct username and password. |
| Post Condition: | Actor has login into the system. |

|  |  |
| --- | --- |
| Use Case: | Logout |
| Summary: | When a admin or agent has done their task or done using the system, they have to logout to avoid other people use to their account. |
| Dependency: | - |
| Actors: | Admin or Agent |
| Pre-condition: | The actor has to login their account. |
| Main Flow: | 1. Actor login to their account. 2. Actor finish using with the system. 3. Actor click the logout button in the system. 4. System will show the page before the actor login to their account if logout successfully. |
| Alternative Flow: | **-** |
| Post Condition: | The actor has logout. |

|  |  |
| --- | --- |
| Use Case: | View schedule |
| Summary: | Actor can use the function to view the available schedule. |
| Dependency: | - |
| Actors: | Admin or Agent |
| Pre-condition: | The actor has to login their account. |
| Main Flow: | 1. Actor login to their account. 2. Actor select the view schedule button. 3. The available schedule showed. |
| Alternative Flow: | **Step 1**: Login failed. |
| Post Condition: | The available schedule showed. |
| Use Case: | View schedule |
| Summary: | Actor can use the function to view the available schedule. |
| Dependency: | - |
| Actors: | Admin or Agent |
| Pre-condition: | The actor has to login their account. |
| Main Flow: | 1. Actor login to their account. 2. Actor select the view schedule button. 3. The available schedule showed. |
| Alternative Flow: | **Step 1**: Login failed. |
| Post Condition: | The available schedule showed. |

|  |  |
| --- | --- |
| Use Case: | Register new agent |
| Summary: | Visitor can create an account as agent. |
| Dependency: | - |
| Actors: | Visitor |
| Pre-condition: | The actor must be at mainpage |
| Main Flow: | 1. Visitor visit mainpage. 2. The form is showed in the front page. 3. Fill in the required information. 4. Submit the form. |
| Alternative Flow: | **Step 1**: Login failed. |
| Post Condition: | The agent account has been created. |

|  |  |
| --- | --- |
| Use Case: | Manage Customers |
| Summary: | Agent manage the customers records |
| Dependency: | - |
| Actors: | Agent |
| Precondition(s): | User has logged in as Agent and click on Manage Customer on Navigation Bar |
| Main Sequence: | 1. User click on create new customer link 2. The web server will redirect user to Add New Customer Page. |
| Alternative Sequence: | **Step 2a:** User click on Edit on one of the customer, the web server will redirect user to Edit Customer Page of the selected customer.  **Step 2b:** User click on Details on one of the customer, the web server will redirect user to the Customer Details Page of the selected customer.  **Step 2c:** User click on Delete on one of the customer, the web server will delete the selected customer records and reload the page. |
| Postcondition(s): | User has been redirected to the New Customer Page. |

|  |  |
| --- | --- |
| Use Case: | Delete Customer |
| Summary: | Agent delete the selected customer record. |
| Dependency: | <<extends>> Manage Customers |
| Actors: | Agent |
| Precondition(s): | User has logged in as Agent and click on Manage Customer on Navigation Bar |
| Main Sequence: | 1. User has clicked on Delete link from a selected customer record. 2. The web server will delete the selected customer record from database and reload the page. |
| Alternative Sequence: | **-** |
| Postcondition(s): | The selected customer record has been removed from the database and the Manage Customer page has been reloaded. |

|  |  |
| --- | --- |
| Use Case: | Add New Customer |
| Summary: | Agent create a new customer record. |
| Dependency: | <<extends>> Manage Customers |
| Actors: | Agent |
| Precondition(s): | User has clicked on Add New Customer link in the Manage Customer Page and redirected to the Add New Customer Page. |
| Main Sequence: | 1. User entered the required information and press Create button. 2. If all the fields are valid, the web server will save the entered information to the database and prompt success message. |
| Alternative Sequence: | **Step 2:** If invalid fields found, the web server will prompt an error message for next attempt. |
| Postcondition(s): | A new record has been stored and prompted a success message. |

|  |  |
| --- | --- |
| Use Case: | Edit Customer |
| Summary: | Agent edit the information of a customer record. |
| Dependency: | <<extends>> Manage Customers |
| Actors: | Agent |
| Precondition(s): | User has clicked on Edit link of a selected customer record in the Manage Customer Page and redirected to the Edit Customer Page. |
| Main Sequence: | 1. User entered the required information and press Create button. 2. If all the fields are valid, the web server will save the entered information to the database and prompt success message. |
| Alternative Sequence: | **Step 2:** If invalid fields found, the web server will prompt an error message for next attempt. |
| Postcondition(s): | A new record has been stored and redirected user to the Manage Customers Page. |

|  |  |
| --- | --- |
| Use Case: | Manage Shipments |
| Summary: | Agent manage the created shipment records |
| Dependency: | - |
| Actors: | Agent |
| Precondition(s): | User has logged in as Agent and click on Manage Shipments on Navigation Bar |
| Main Sequence: | 1. User click on Book New Shipment link 2. The web server will redirect user to Booking Page. |
| Alternative Sequence: | **Step 2a:** User click on Details on one of the shipments, the web server will redirect user to the Shipment Details Page of the selected shipment.  **Step 2b:** User click on Delete on one of the shipments, the web server will delete the selected shipment record and reload the page. |
| Postcondition(s): | User has been redirected to the Booking Page. |

|  |  |
| --- | --- |
| Use Case: | View Shipment Details |
| Summary: | Agent view the details of a shipment record |
| Dependency: | <<extends>> Manage Shipments |
| Actors: | Agent |
| Precondition(s): | User has logged in as Agent and click on Manage Shipments on Navigation Bar |
| Main Sequence: | 1. User click on Details link on a shipment record. 2. The web server will redirect user to Shipment Details Page of the selected shipment record. |
| Alternative Sequence: | **-** |
| Postcondition(s): | User has been redirected to the Shipment Details Page of the selected shipment record. |

|  |  |
| --- | --- |
| Use Case: | Delete Shipment |
| Summary: | Agent delete the selected shipment record. |
| Dependency: | <<extends>> Manage Shipments |
| Actors: | Agent |
| Precondition(s): | User has logged in as Agent and click on Manage Shipments on Navigation Bar |
| Main Sequence: | 1. User has clicked on Delete link from a selected shipment record. 2. The web server will delete the selected shipment record from database and reload the page. |
| Alternative Sequence: | **-** |
| Postcondition(s): | The selected shipment record has been removed from the database and the Manage Shipments page has been reloaded. |

|  |  |
| --- | --- |
| Use Case: | Book New Shipment |
| Summary: | Agent creates a new shipment record |
| Dependency: | - |
| Actors: | Agent |
| Precondition(s): | User has logged in as Agent, click on Booking on Navigation Bar and enters the Booking Page |
| Main Sequence: | 1. User select the available shipping schedule by clicking Select link on the desired shipping schedule. 2. The web server will redirect user to New Shipment Page with the Shipping Schedule selected. 3. The user entered the required fields and click Submit button. 4. If all the fields are valid, the web server will save the entered information into the database server and redirect the user back to the Agent main page. |
| Alternative Sequence: | **Step 3:** If invalid fields found, the web server will prompt an error message for next attempt. |
| Postcondition(s): | A new shipment record has been successfully created and stored into the database server. The user has been successfully redirected to the Agent main page. |

|  |  |
| --- | --- |
| Use Case: | Manage Vessel |
| Summary: | Agent manage the created shipment records |
| Dependency: | - |
| Actors: | Admin |
| Precondition(s): | User has logged in as Admin and click on Manage Vessel on Navigation Bar |
| Main Sequence: | 1. User click on Create New Vessel 2. The web server will redirect user to Booking Page. |
| Alternative Sequence: | **Step 2:** User click on Delete on one of the vessel, the web server will delete the selected vessel record and reload the page. |
| Postcondition(s): | User has been redirected to the Add New Vessel Page. |

|  |  |
| --- | --- |
| Use Case: | Create New Vessel |
| Summary: | Admin create a new customer record. |
| Dependency: | <<extends>> Manage Vessel |
| Actors: | Admin |
| Precondition(s): | User has clicked on Create New Vessel link in the Manage Vessel Page and redirected to the Add New Vessel Page. |
| Main Sequence: | 1. User entered the required information and press Create button. 2. If all the fields are valid, the web server will save the entered information to the database and prompt success message. |
| Alternative Sequence: | **Step 2:** If invalid fields found, the web server will prompt an error message for next attempt. |
| Postcondition(s): | A new record has been stored and prompted a success message. |

|  |  |
| --- | --- |
| Use Case: | Delete Vessel |
| Summary: | Admin delete the selected vessel record. |
| Dependency: | <<extends>> Manage Vessel |
| Actors: | Admin |
| Precondition(s): | User has logged in as Admin and click on Manage Vessel on Navigation Bar |
| Main Sequence: | 1. User has clicked on Delete link from a selected vessel record. 2. The web server will delete the selected vessel record from database and reload the page. |
| Alternative Sequence: | **-** |
| Postcondition(s): | The selected vessel record has been removed from the database and the Manage Vessel page has been reloaded. |

|  |  |
| --- | --- |
| Use Case: | Manage Agents |
| Summary: | Admin manage the agent records |
| Dependency: | - |
| Actors: | Admin |
| Precondition(s): | User has logged in as Admin into the Admin main page. |
| Main Sequence: | 1. User click on Manage Agents on Navigation Bar. 2. The web server will redirect user to Manage Agents Page. |
| Alternative Sequence: | **-** |
| Postcondition(s): | User has been redirected to the Manage Agents Page. |

|  |  |
| --- | --- |
| Use Case: | Approve Registration |
| Summary: | Admin approve the selected agent registration record. |
| Dependency: | <<extends>> Manage Agents |
| Actors: | Admin |
| Precondition(s): | User has logged in as Admin and click on Manage Agents on Navigation Bar |
| Main Sequence: | 1. User has clicked on Approve link from a selected agent record in the New Registration to be Approved table. 2. The web server will update the registration status the selected agent record in database and reload the page. |
| Alternative Sequence: | **-** |
| Postcondition(s): | The registration status of the selected agent record has been updated from the database and the Manage Agents page has been reloaded. |

|  |  |
| --- | --- |
| Use Case: | Manage Shipping Schedule |
| Summary: | Admin manage the shipping schedule records |
| Dependency: | - |
| Actors: | Admin |
| Precondition(s): | User has logged in as Admin |
| Main Sequence: | 1. User click on Manage Shipping Schedule on Navigation Bar 2. The web server will redirect user to Manage Shipping Schedule Page. |
| Alternative Sequence: | **-** |
| Postcondition(s): | User has been redirected to the Manage Shipping Schedule Page. |

|  |  |
| --- | --- |
| Use Case: | Add New Shipping Schedule |
| Summary: | Admin create a new shipping schedule record. |
| Dependency: | <<extends>> Manage Shipping Schedule |
| Actors: | Admin |
| Precondition(s): | User has clicked on Add New Shipping Schedule link in the Manage Shipping Schedule Page and redirected to the Add New Schedule Page. |
| Main Sequence: | 1. User entered the required information and press Create button. 2. If all the fields are valid, the web server will save the entered information to the database and prompt success message. |
| Alternative Sequence: | **Step 2:** If invalid fields found, the web server will prompt an error message for next attempt. |
| Postcondition(s): | A new record has been stored and prompted a success message. |

### Sequence diagram

#### Login sequence diagram

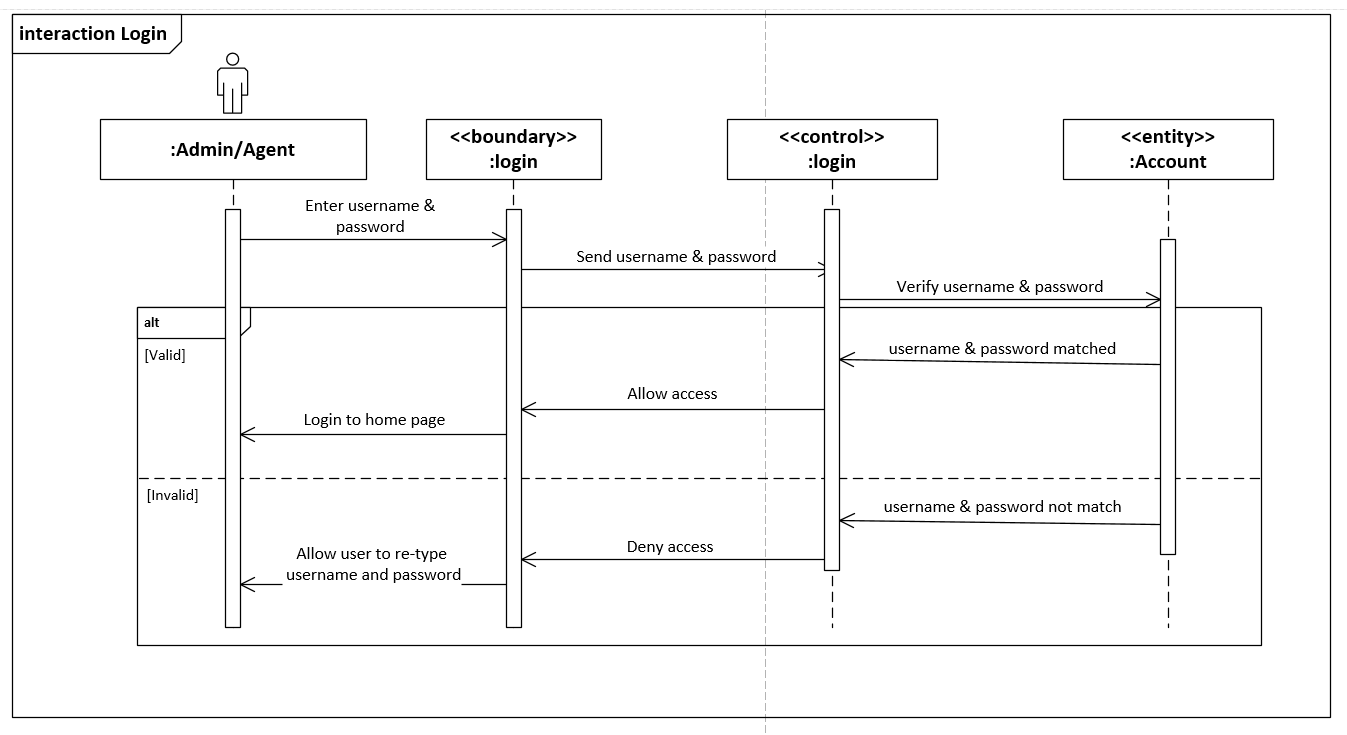


Figure 2: Login Sequence Diagram

#### Logout sequence diagram

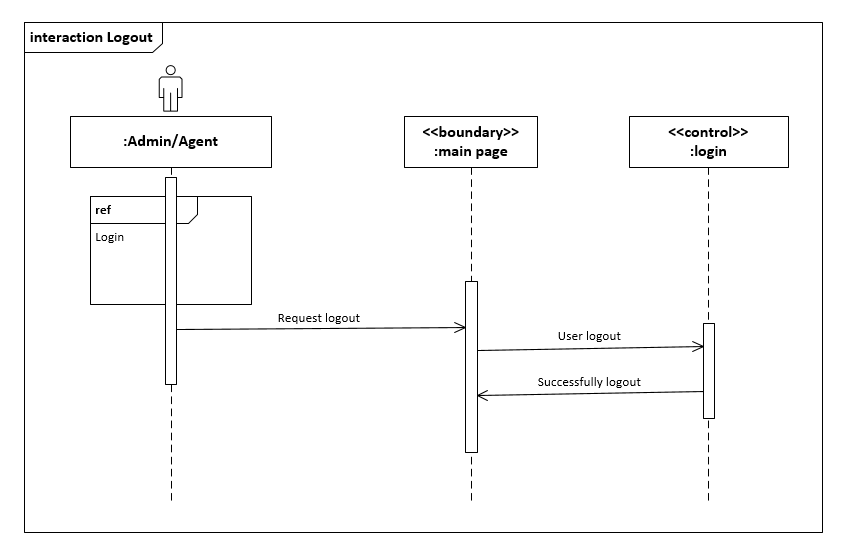


Figure 3: Logout Sequence Diagram

#### Register as new Agent sequence diagram

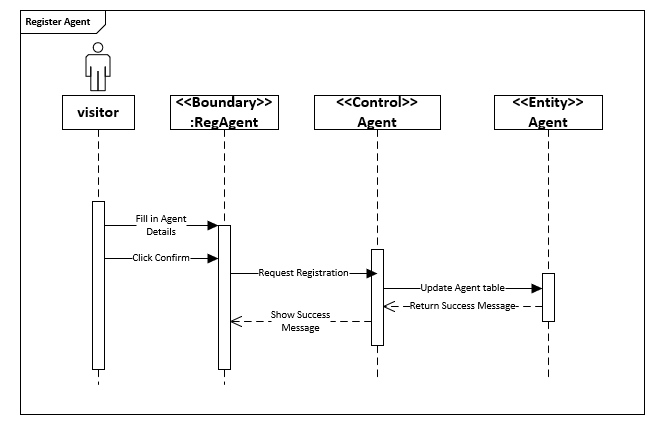


Figure 4: Register agent sequence diagram

#### Register new customer sequence diagram sequence diagram

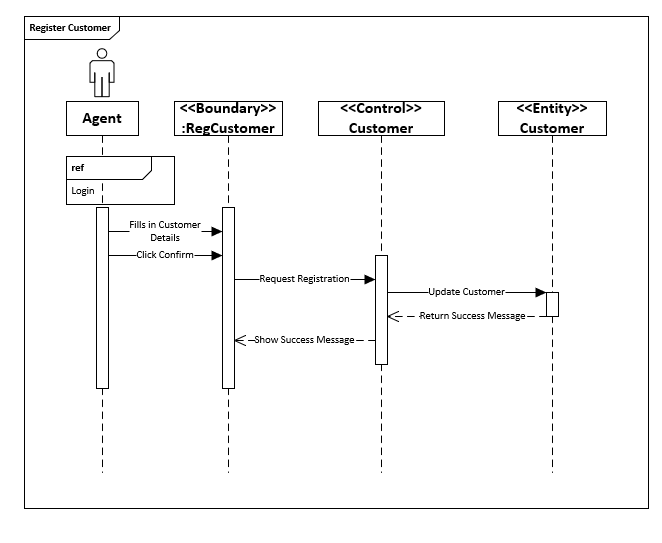


Figure 5:Register new customer sequence diagram

#### View schedule sequence diagram

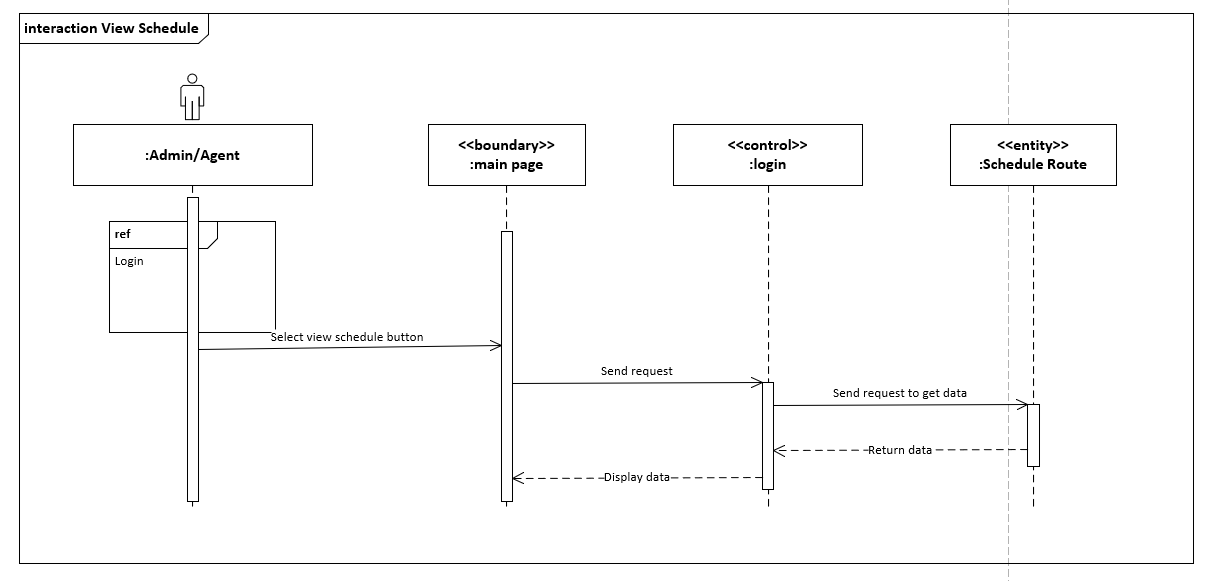


Figure 6: View schedule sequence diagram

#### Book new shipment sequence diagram

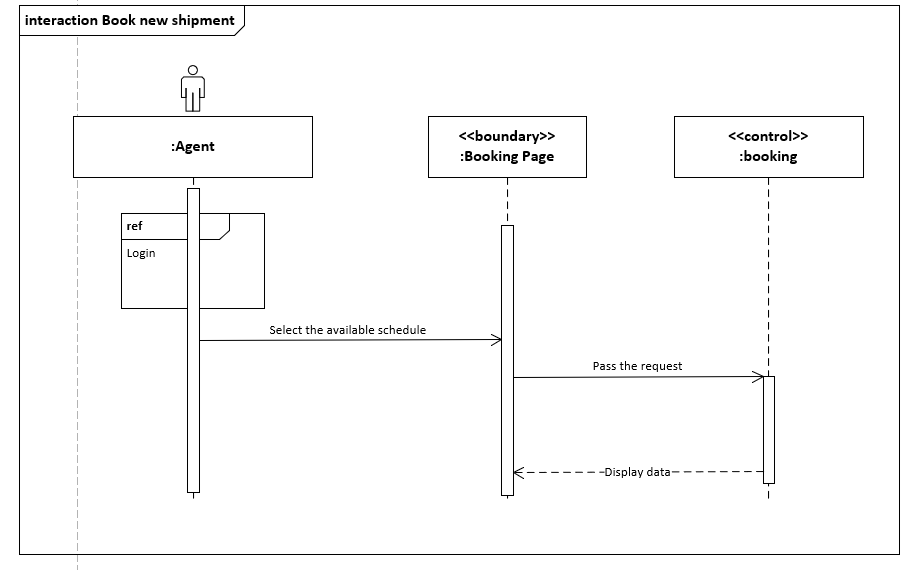


Figure 7: Book new shipment sequence diagram

#### Manage Customer

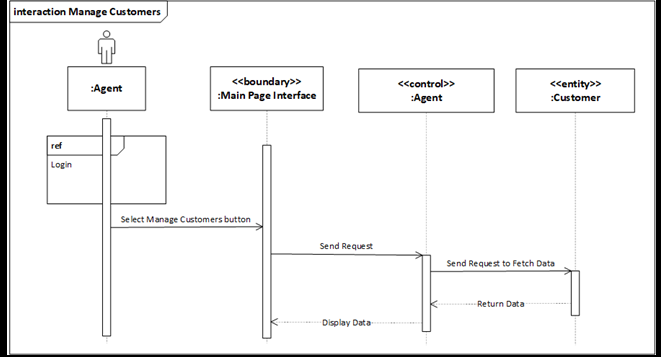


Figure 8:Manage Customer

#### Delete Customer

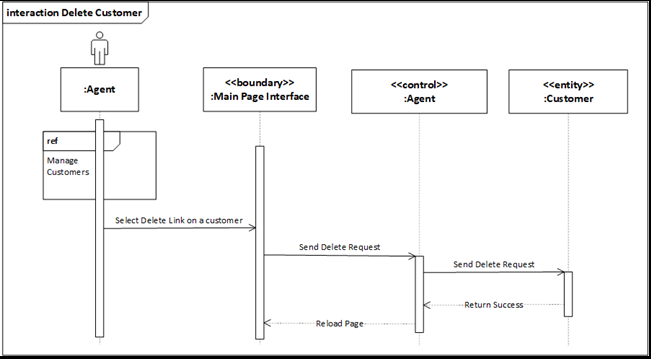


Figure 9: Delete Customer

#### Edit Customer

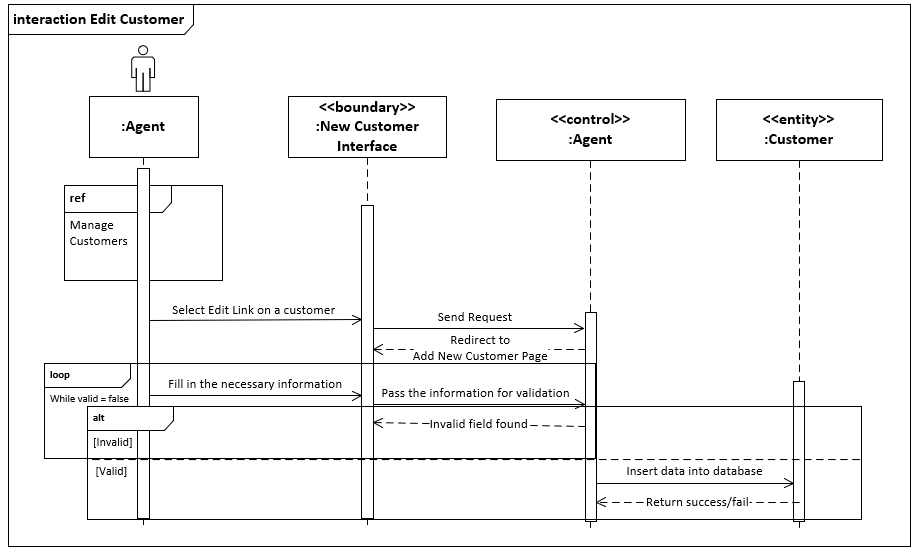


Figure 10: Edit Customer Sequence Diagram

#### Delete Shipment

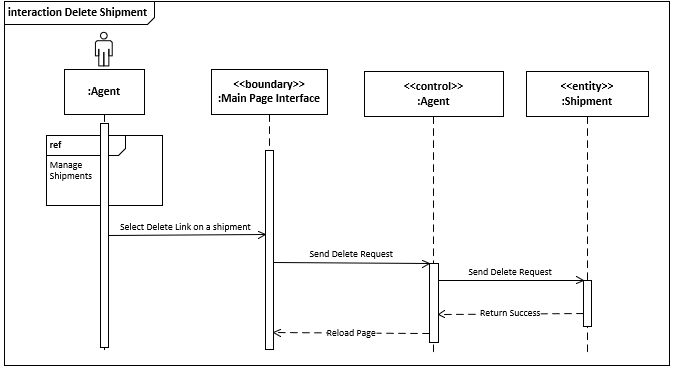


Figure 11:Delete Shipment

#### Approve new Agent Registration

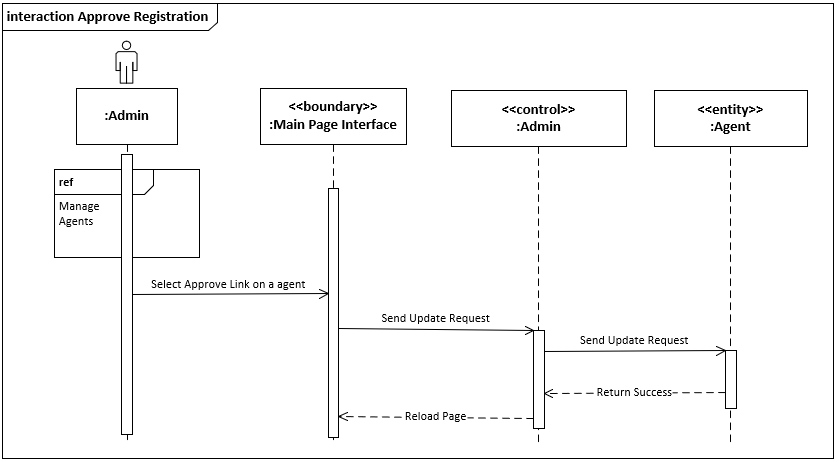


Figure 12: Approve new agent registration sequence diagram

#### Add Vessel

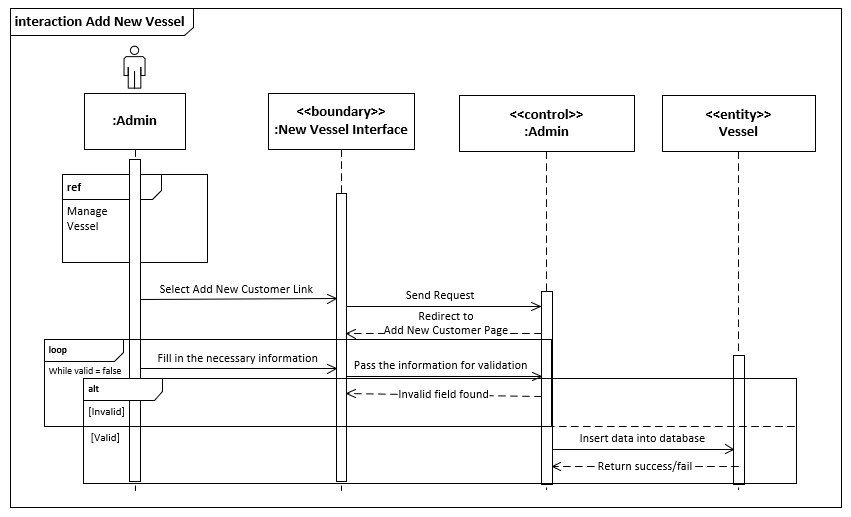


Figure 13: Add new vessel sequence diagram

### Class diagram

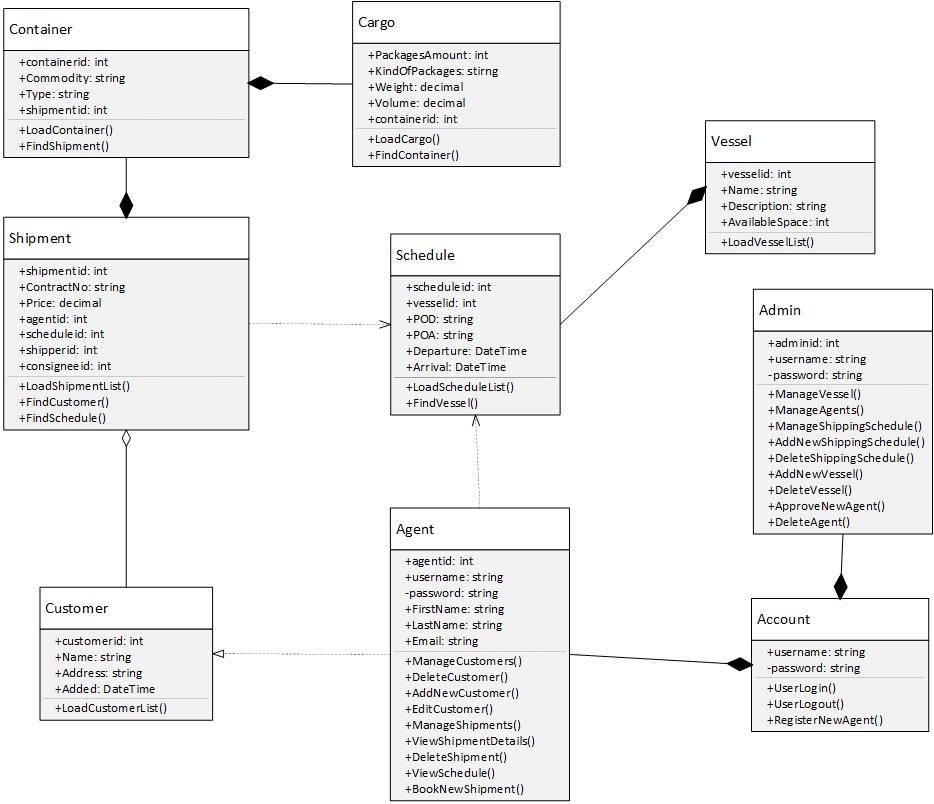
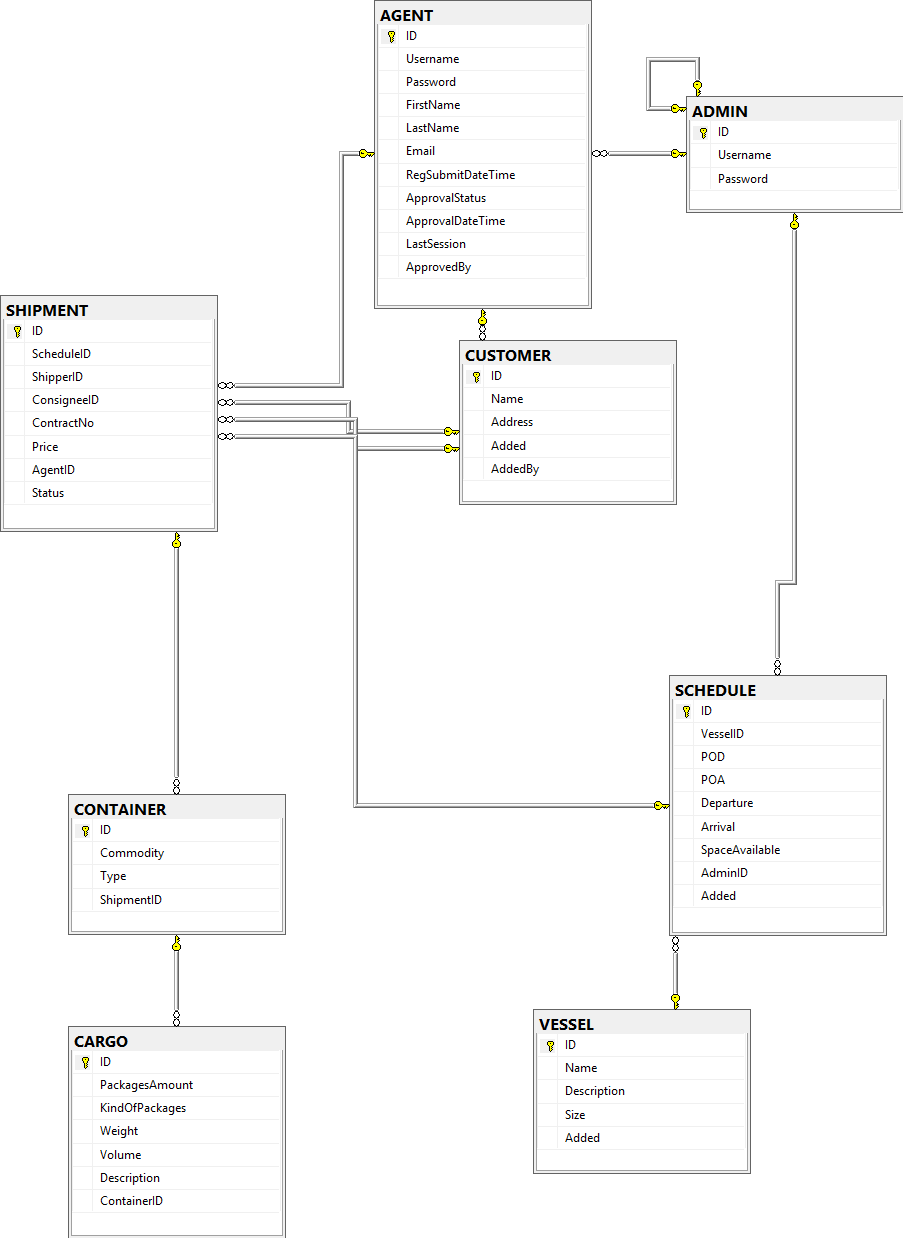


Figure 14:Class diagram

### ERD Diagram



## Cloud Architectural Diagram

The initial Maersk Line Container Management Web Application Service implementation on cloud environment will be mainly focusing on South East Asia region, which the developer will make decision pf placing the main web application in South East Asia. Due to the database first method is applied throughout the development, the server Microsoft SQL will be shared to the main server.

For a better implementation, the developer has decided there is a default of two instances to be scaled up and down. Some of the service subscribed in the web application is the web service to provide the container for holding Maersk Line Container Management Web Application Service with two instances to be scaled out and in according to server performance needs and wants. Beside that, there is a traffic manager that will be monitoring the connection between client and server to decide which is the best route between them. Lastly, the Azure database for Microsoft SQL is another important features acquired to hold the necessary data for daily operation.

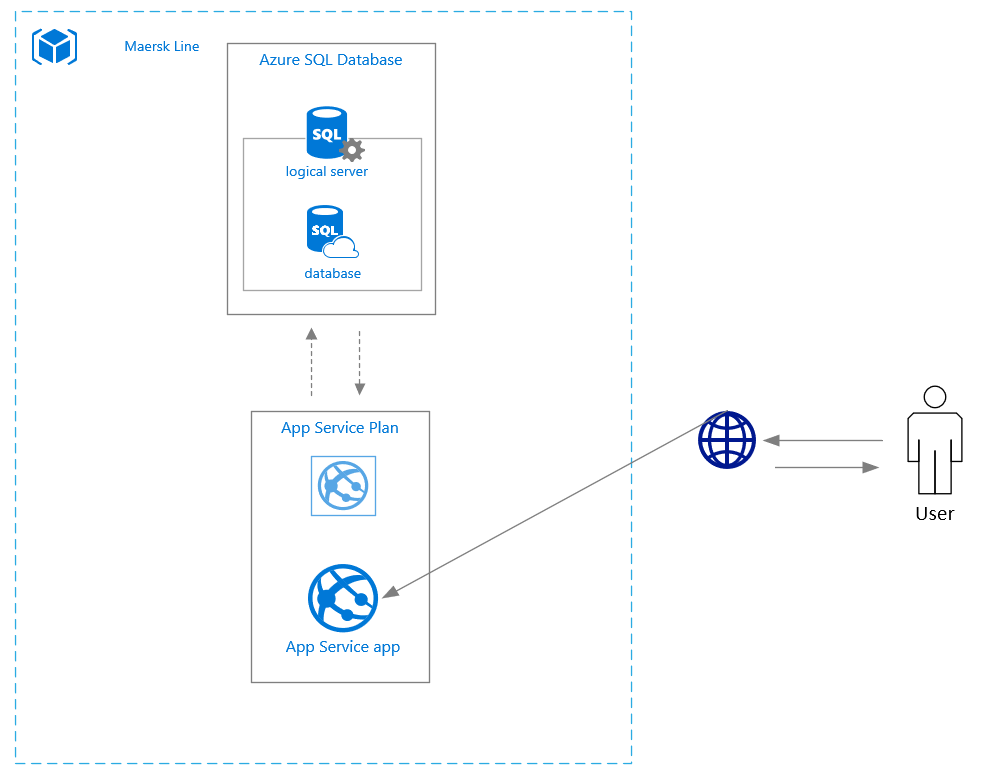


Figure 15: Cloud Architectural Diagram

In conclusion, the advantage of this design is to prevent the developer to create additional logic to manage failover in the region resources as the same web application is deployed to the regions without any region-specific configuration. Also, Application performance is not impacted by failover because the web application and the databases are always co-located.

# Implementation

## Application Development

The development of Maersk Line Container Management application is done using with IDE, Visual studio 2017 and MVC structure was used to develop the user interface which mainly broken down into three component, Model, Controller and View. Other than that, Microsoft SQL database is integrated to store and retrieve the application data based on user needs. Following figure below is the file structure.

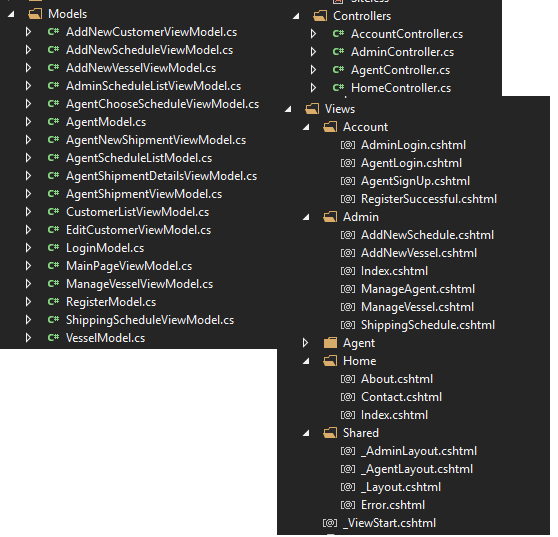


Figure 16: MVC File Structure

Controllers, which is the recommended location for controllers. The MVC framework requires the names of all controllers to end with "Controller", such as HomeController, LoginController, or ProductController. Models, which is provided for classes that represent the application model for your MVC Web application. This folder usually includes code that defines objects and that defines the logic for interaction with the data store. Typically, the actual model objects will be in separate class libraries. However, when you create a new application, you might put classes here and then move them into separate class libraries at a later point in the development cycle. Views, which is the recommended location for views. Views use ViewPage (.aspx), ViewUserControl (.ascx), and ViewMasterPage (.master) files, in addition to any other files that are related to rendering views. The Views folder contains a folder for each controller; the folder is named with the controller-name prefix. For example, if you have a controller named HomeController, the Views folder contains a folder named Home. By default, when the ASP.NET MVC framework loads a view, it looks for a ViewPage (.aspx) file that has the requested view name in the Views\*controllerName* folder. By default, there is also a folder named Shared in the Views folder, which does not correspond to any controller. The Shared folder is used for views that are shared across multiple controllers. For example, you can put the Web application's master page in the Shared folder. The first page will be displayed is the index.cshtml in View which shows all kinds of basic function to be accessed by specific user and request will be redirected to proper controller with given page variable and homepage will be displayed by default if specific page is not mentioned. The application UI is developed with the combination of HTML and C#. The following below is the homepage for Maersk Line Container Management Web Application.

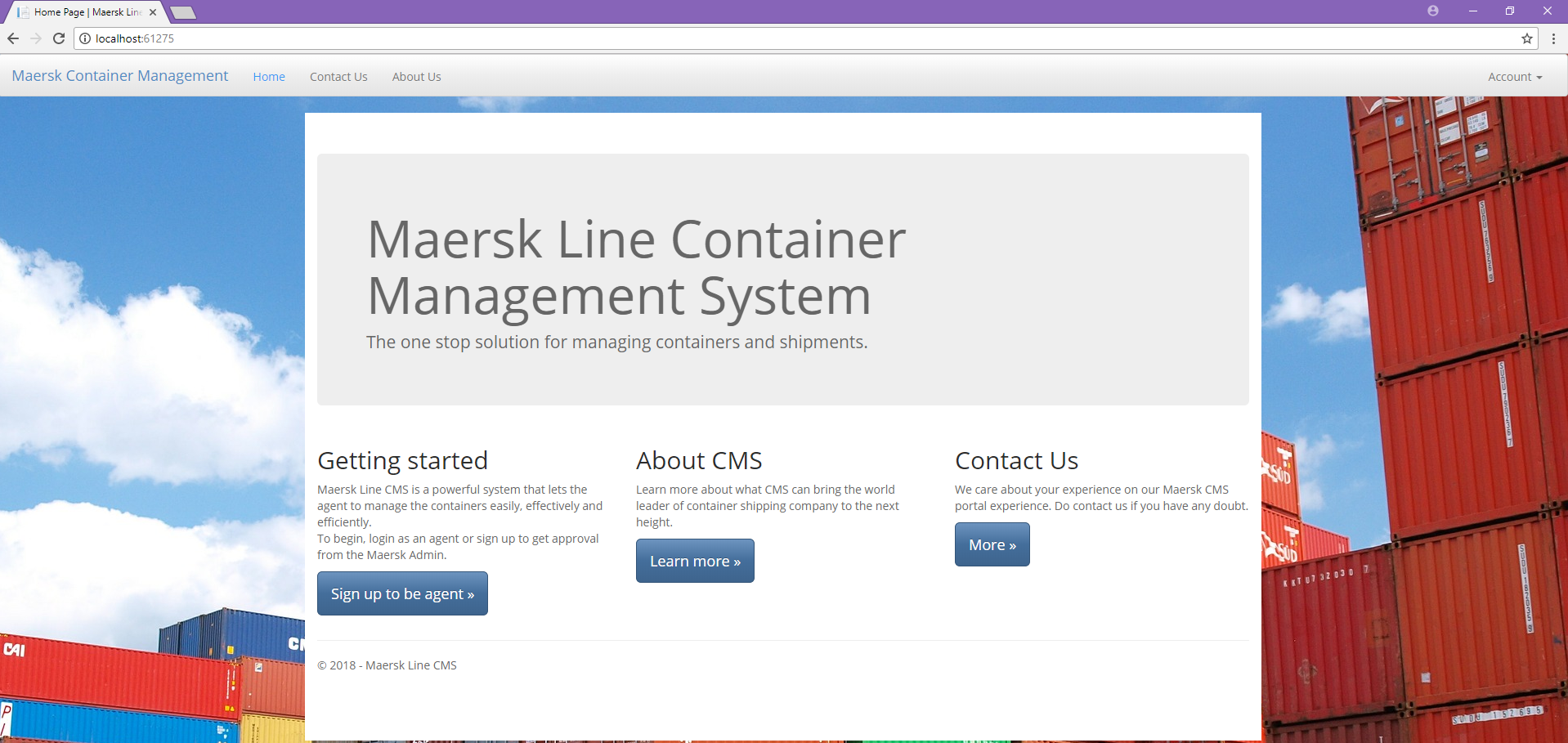


Figure 17: Maersk Line Container Management

In the end, the version control of application throughout the development is done with assisting of a tool called GitHub which the final output can be viewed with this link, https://github.com/terranleong/UIA\_FlightBookingWebApplication.

## Azure Publishing

The web application will be hosted on Microsoft Azure. There are several steps that needed to be carried out to fully deploy the web application and database to Microsoft Azure. However, the selection of programming language and database will affect the process to deploy the application and database to Microsoft Azure. The following shows the steps of deploying web application and database to Microsoft Azure.

### App service publish

Step 1- Create new web service on Azure

Firstly, create a new resource group that named “MAERSKLINE” and located at Southeast Asia. The figure bellows shows the creation of resource group.

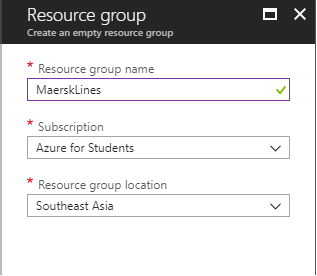


Figure 18: Resource Group create

After that, create a new web app which requires the user to register app name, choose subscription, resource group, app service plan, location and operating system. Choose a most appropriate subscription that suit the web application with current situation. Also, choose a most suitable location to host the web application. The figure below shows the creation of app service.

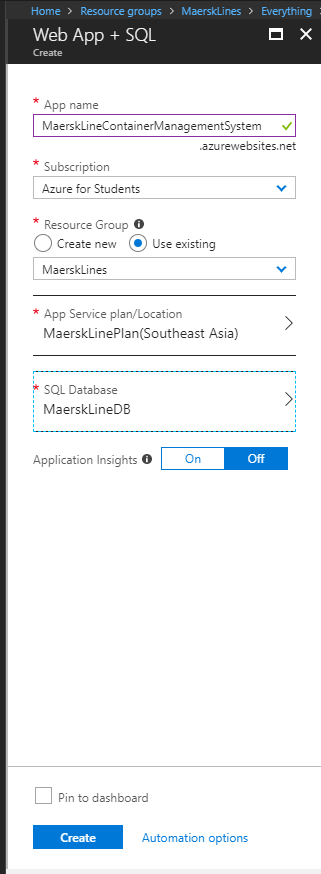


Figure 19: App service Registration

Step 2 –Publish Web Application

Step 2 is Open Visual Studio and open the application’s project. Find the main project in solution explorer. Right click the main project and press “Publish”. The following shows the snapshot of this process.

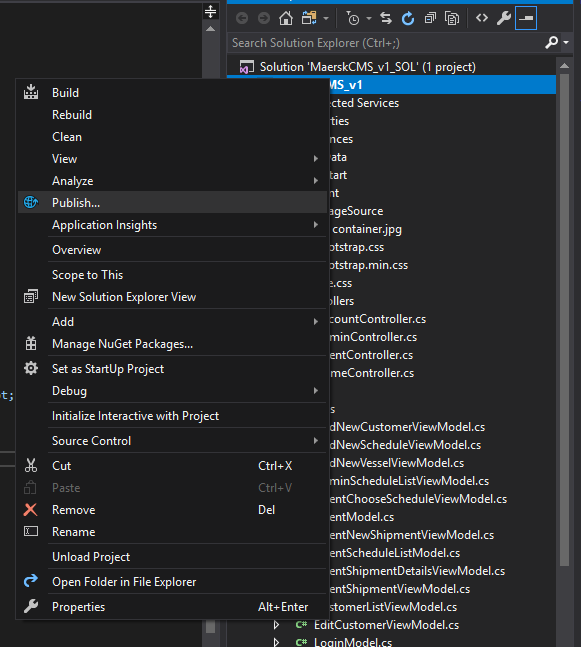


Figure 20: Publish Visual Studio

After that, a window will pop-out and it requires user to login to the Microsoft Azure account. Choose the correct subscription, then choose, “Azure for Students”. A list of resource groups will be listed and choose the correct one and press “OK” button to publish the web application.

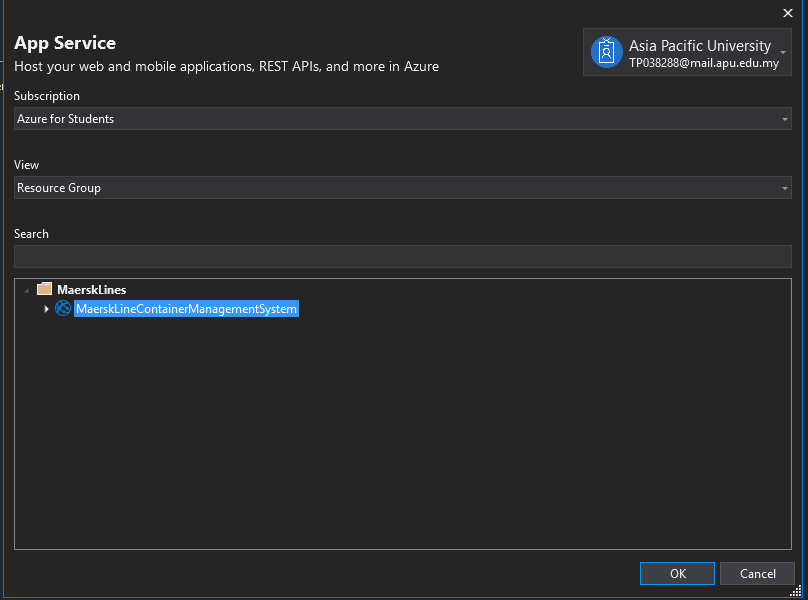


Figure 21: App Service Publish

Then wait for the process to complete, then the following message will show say is done.

### Database publish

Step 1 – Create New Web Service on Azure

Firstly, create a resource group for database which named “MAERKLINECONTAINERMANAGEMENTSYSTEM”. Then, create a new database service which is “SQL Database”. Fill up all the required information of the creation of new SQL Database service. The following figure shows the creation of new SQL Database service.

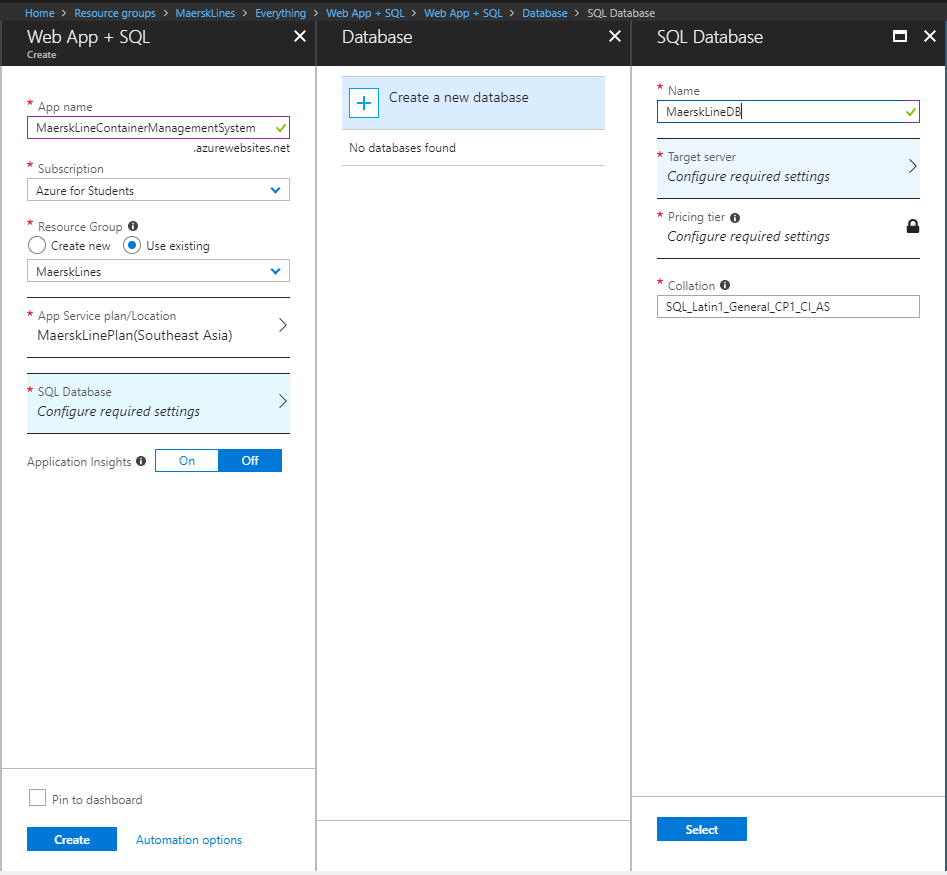


Figure 22:Database configuration

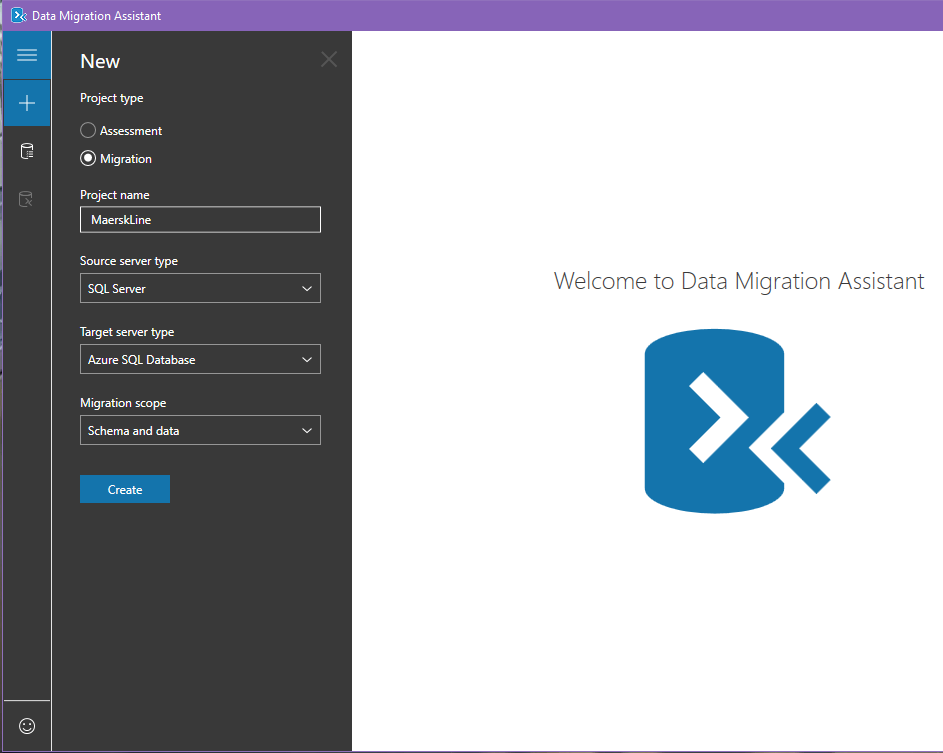
Then after create the SQL database service, install an application called “Microsoft Data Migration Assistant (DMA)” to migrate the database from local server to Azure. Open DMA and press create to create a new migration project. Fill up all required information and press create button. The following figure shows the configuration of migration project

Figure 23: create of migration DB

Insert the server name of local server so that the database can be retrieved and migrated. Choose the correct database that needs to be migrated to Azure. In this case, “DDAC” is chosen to be migrated. Press “Next” button to proceed to next steps.

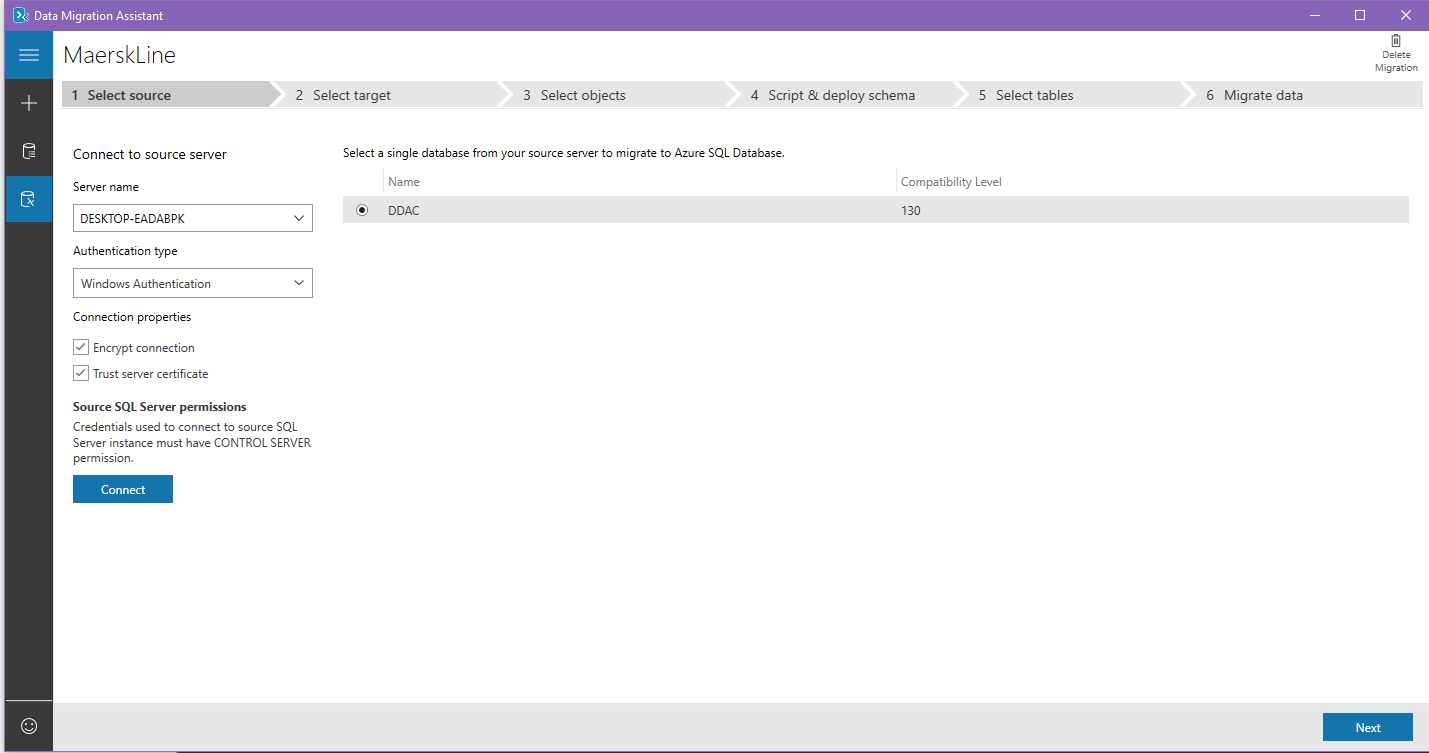


Figure 24: Select DB

Then insert name of the server name of the SQL Database in Azure and connect to the server. It requires the username and password to access the SQL Server and this username and password is configured in previous step. Choose the Database as the destination of the migration. In this case, the database called “MAERKDB”. Press “Next” button to proceed to next step.

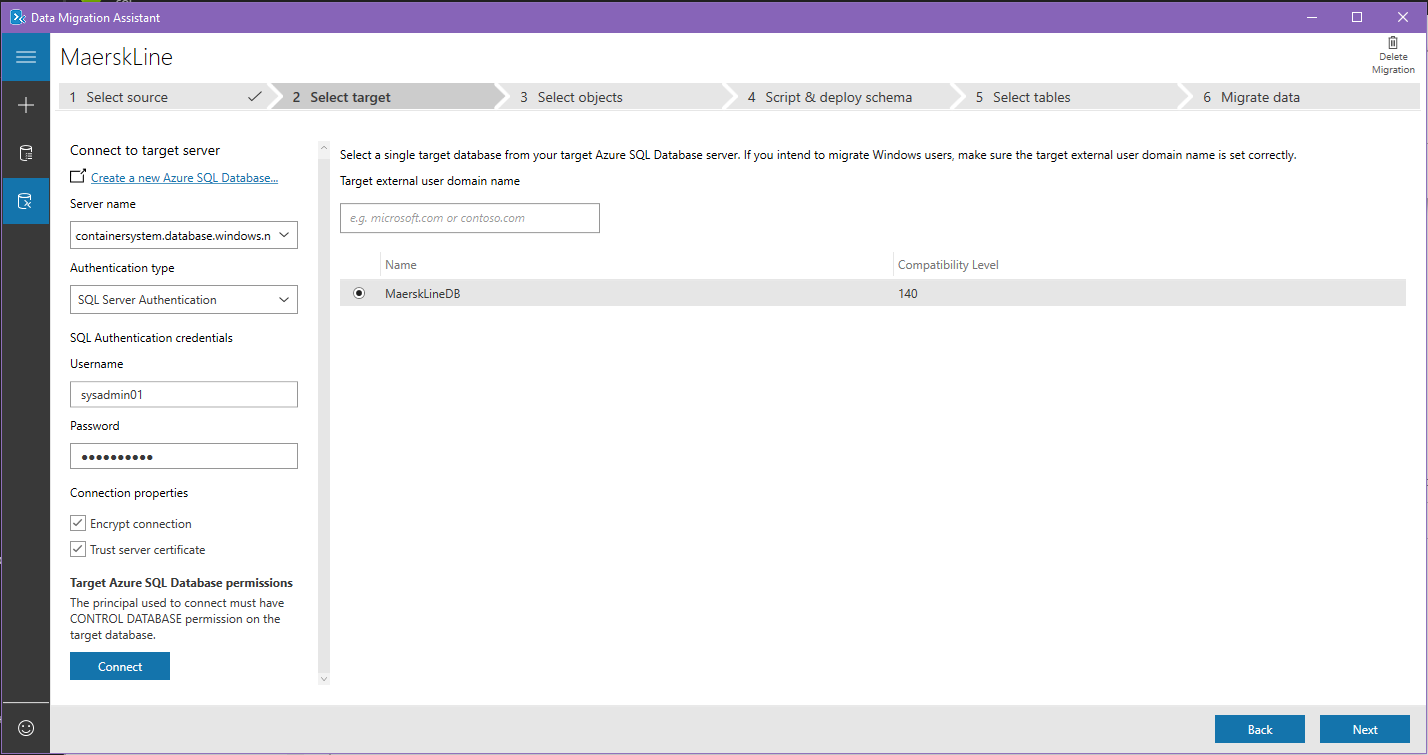


Figure 25: Select Target to migrate

Select the table that needs to be migrated to the Azure. Then press“Generate SQL Script” to proceed to the next step. The following figure shows the table that needs to be migrated to Azure.

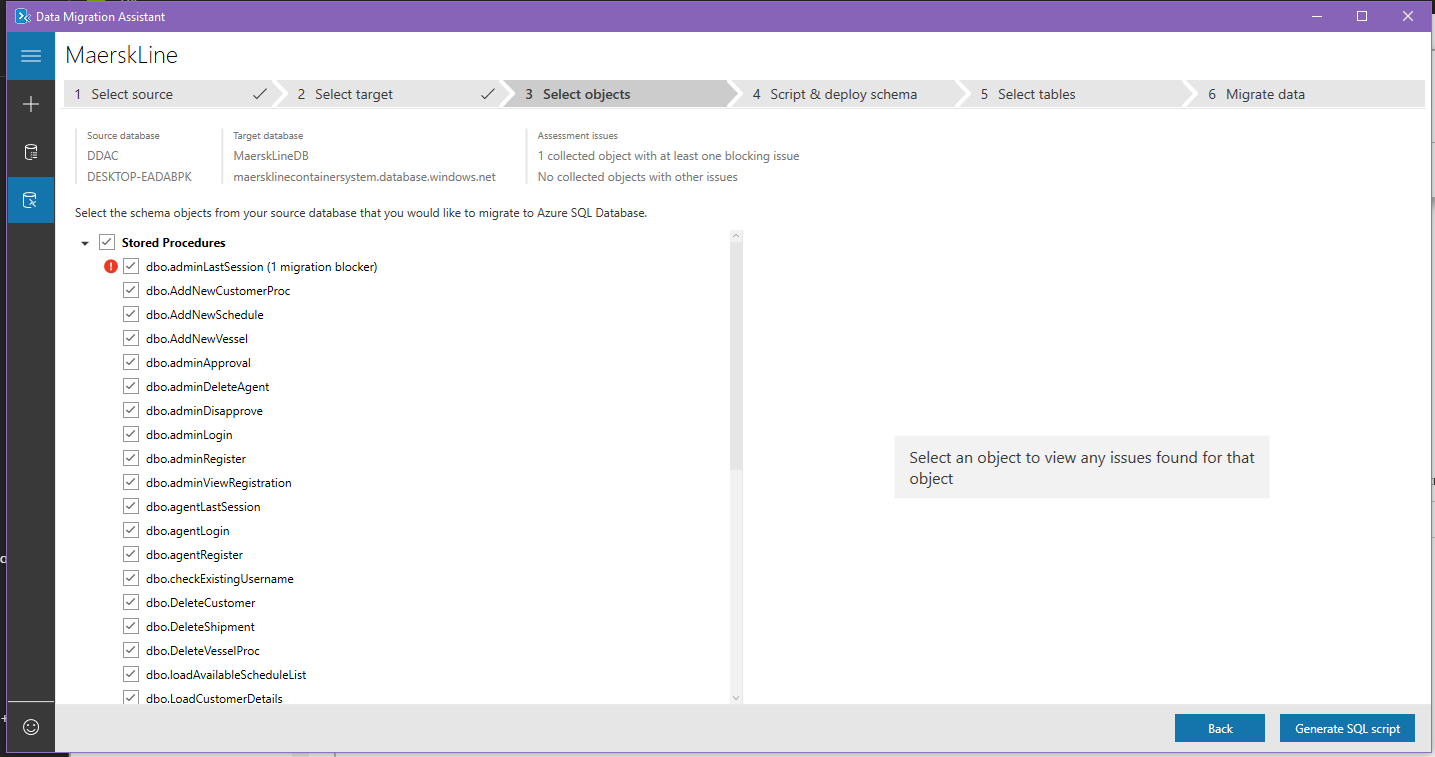


Figure 26:Generate SQL Script

The system will automatically generate the SQL script that re-create the whole database on Azure. If there is no any bug or error, then proceed and press “Deploy Schema” to migrate the database to Azure. The following figure shows the migration script of the database in this system.

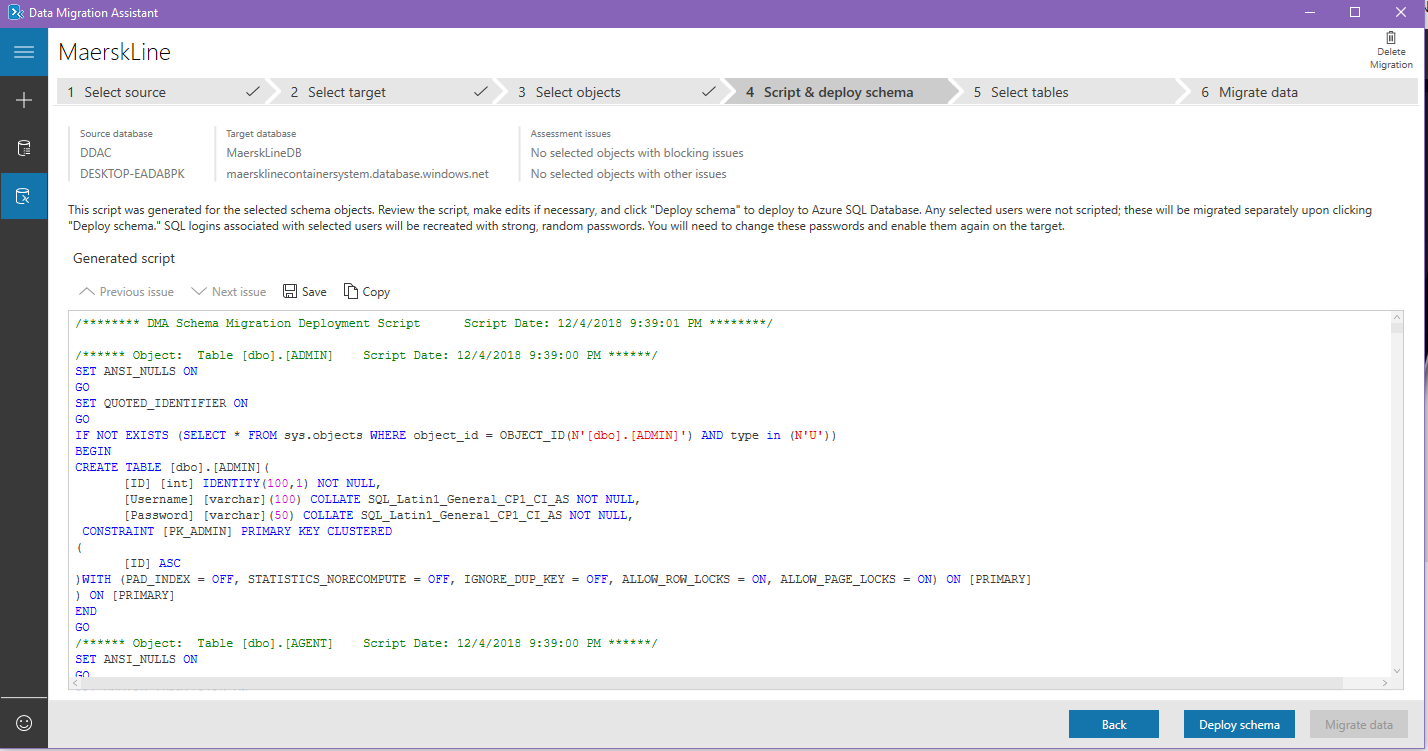


Figure 27:Depoly schema

The DMA will execute all the query and ready for the migration of database to Azure. Press “Migrate Data” to proceed to next step.

This part here the DMA will display the status of the table whether it is ready to be migrated to Azure. Then tick all the table and press the button of “Start data migration” to complete migration process.

## Application Scaling

Based on the (Microsoft Azure, 2017) Azure could environment does provide the web service for auto scaling which the resource can be powerful based on the needs to match with performance requirement. This is quite important when Maersk Line Container Management Web Application increase the workload, more resource is required to maintain the needed performance levels and satisfaction level agreements. Extra resources allocated during the top peak will not be a waste as it will automatically deallocated when no longer needed to minimize the operation cost. It has the benefit of not having a dedicated personal to monitor the server performance and deciding which to proceed with allocating or deallocating or more resource.

### SQL Database Scaling

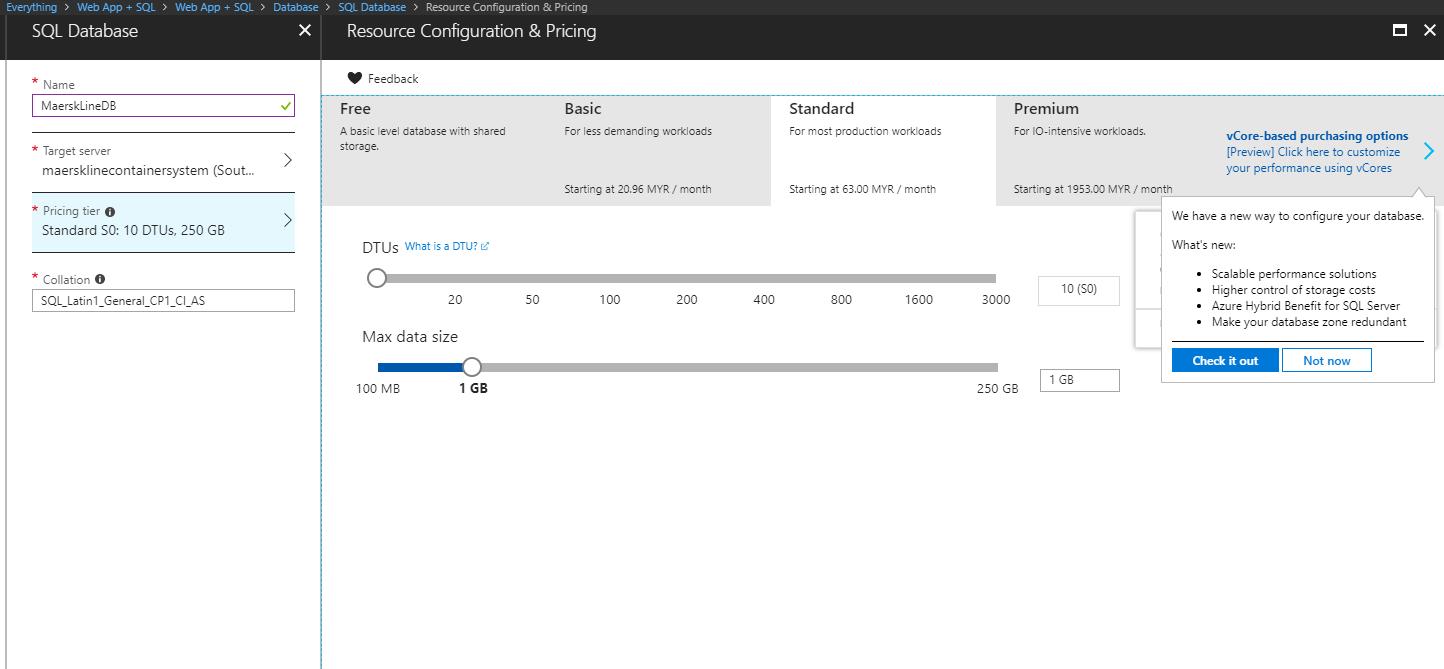
A standard plan for the SQL database will be use for initial deployment phase which do not need much storage and computation. In the current situation, there 10 DTUs and 500MB provided for storage size and this is sufficient enough for the deployment testing.

Figure 28: Scaling Azure

The figure above shows the current scale of the SQL database in Azure. The normal or basic plan of the Azure SQL database service is not suitable for this system, due to the maximum amount of database size is only 2GB and only 5DTUs are provided. With that size, it is not enough for the system to handle the large amount of data from different countries and the performance of the database will slow down. As for the current state, the standard plan is a better plan to be utilize for initial deployment stage as it is slowly deploying to other country(Microsoft Azure, 2018).

Businesses grow, more data and DTU will be needed to be assign to the system, but it depends on the situation. If current plan is not sufficient to support the bigger load of the system, then a bigger plan such as Premium tier will be needed in the future in order to handle the greater amount of transaction.

### Web application scaling

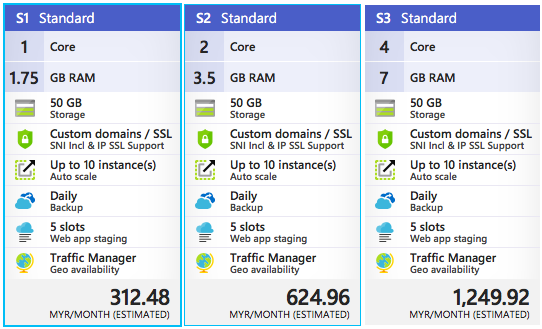


Figure 29:App service plan

For Maersk Line Container management system web application, S1 standard pricing tier will be chosen to host the web application. The standard pricing tier, the storage given for the user is 50GB, compare to the basic pricing tier is only 10GB storage for the user(Microsoft Azure, 2018). Standard plan is much reliable and scalable as the system might be improve in the future and the size of the web application will increase(Microsoft Azure, 2018). With that being said, the basic plan is not scalable, thus, the standard plan provide auto scaling for the user while standard plan provide manual scaling. Auto scaling is very important for the app service will automatically scale up or down the number of virtual machine’s instances based on average performance of the CPU. Different than the basic plan, the user has to manually adjust the number of instances.

Apart from that, in the standard plan, the traffic manager is also provide for the user while basic plan do not provide traffic manager(Microsoft Azure, 2018). Traffic manager is very important feature that helps the user to route to the better performed internet location based on the available endpoint. This could reduce the latency of using this web application. Furthermore, standard plan allow the Maersk Line to use their own domain name instead of the default Azure’s domain name. Thus, standard plan also gives the feature of the daily backup for the system and 5 slots of web application staging. The benefit of deploying the web application in slot is to validate the system changes in a staging development before swapping it with the development slot. Which allow the developer to test the system first hand in other slot.

## Reliability and performance

First and for most, the reliability improvement is done through deploying the application on three or more servers in different parts of South East Asia. In this case, the developer must deploy the application in the central of South East Asia. This can improve the respond time and throughout, so the user can enjoy the full experience when browsing the Maersk Container Management website with the help of traffic manager.

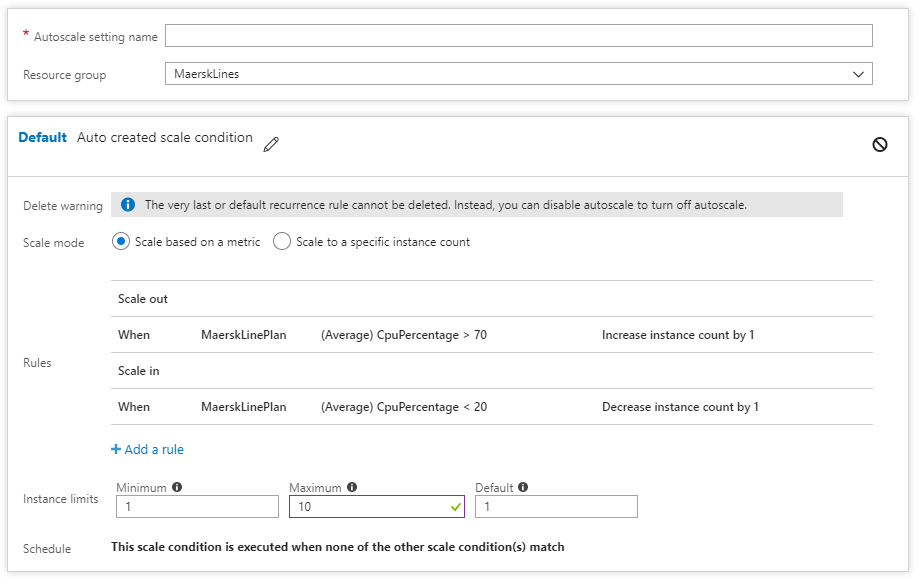
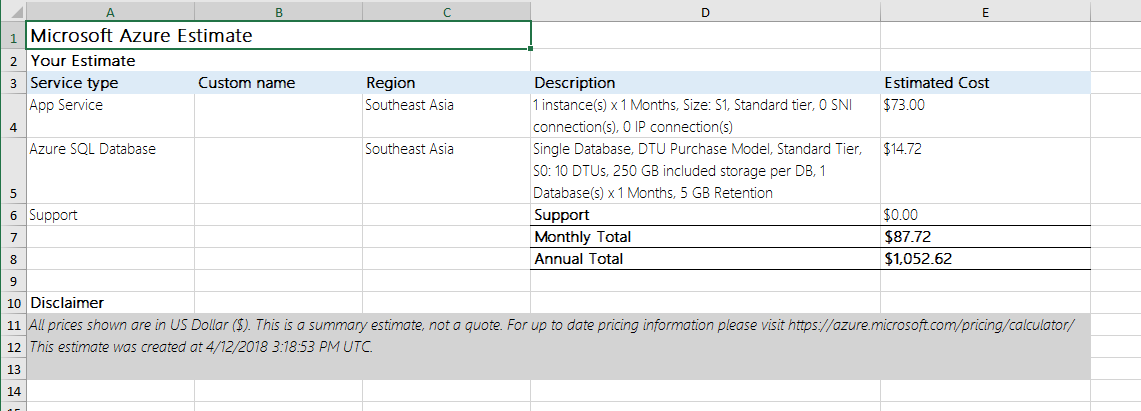


Figure 30: Auto Scaling

Beside that, the SQL database service also provide a feature called geo-replication, the benefit of it is that the availability and reliability of the database. This feature allow the user to configure up to four readable secondary database in the same or different regions. Secondary database is available for querying and for failover if the primary database fails or inability to access it(Microsoft Azure, 2018). Beside that, the recovery of the primary database can be done by retrieving the data from the secondary database as well. Thus, the corrupted data in primary database, primary database can be taken offline and failover to secondary database and also perform recovery of primary database by restoring data from secondary database. Since these are the features that provided by Azure in S1 tier service plan, in the future the business expend more and more Maersk Line can use it to deploy on different region(Microsoft Azure, 2018).

The total cost incurred by implementing the architecture above is down in the table below:



# Test Plan & Testing Discussion

## Test Plan

### Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Parameter | Test Description | Expected Result | Actual Result | Status |
| TCD-001 | Username | 1. Do not enter any value 2. Press Login Button | Display error message | Display Error message” Username & password field is require” | Pass |
| TCD-002 | Password | 1. Do not enter any value 2. Press Login Button | Display error message | Display Error message” Username & password field is require” | Pass |
| TCD-003 | Username, Password | 1. Do not enter any value 2. Press Login Button | Display error message | Display Error message” Username & password field is require” | Pass |
| TCD-004 | Admin Account | 1. Enter username 2. Enter password 3. Press Login Button | Display Admin page | Display admin page | Pass |
| TCD-005 | Agent Account | 1. Enter username 2. Enter Password 3. Press Login Button | Display Agent page | Display Agent page | Pass |

### Register as admin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Parameter | Test Description | Expected Result | Actual Result | Status |
| TCD-006 | Register button | 1. Press Register button | Display register page | Display register page | Pass |
| TCD-007 | FirstName | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display Error message “Please fill in FirstName” | Pass |
| TCD-008 | LastName | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display Error message “Please fill in LastName” | Pass |
| TCD-009 | Username | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display error message “Please fill in username” | Pass |
| TCD-010 | Password | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display error message “Please fill in password” | Pass |
| TCD-011 | Confirm Password | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display error message “Please fill in confirm password” | Pass |
| TCD-012 | EmailAddress | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display error message “Please fill in confirm password” | Pass |

### Register as Agent

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Parameter | Test Description | Expected Result | Actual Result | Status |
| TCD-006 | Register button | 1. Press Register button | Display register page | Display register page | Pass |
| TCD-007 | FirstName | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display Error message “Please fill in FirstName” | Pass |
| TCD-008 | LastName | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display Error message “Please fill in LastName” | Pass |
| TCD-009 | Username | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display error message “Please fill in username” | Pass |
| TCD-010 | Password | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display error message “Please fill in password” | Pass |
| TCD-011 | Confirm Password | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display error message “Please fill in confirm password” | Pass |
| TCD-012 | EmailAddress | 1. Do not enter any value 2. Press Sign Up button | Display Error message | Display error message “Please fill in confirm password” | Pass |

### Add New Shipping Schedule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Parameter | Test Description | Expected Result | Actual Result | Status |
| TCD-0013 | Add new shipping schedule | 1. Click add new shipping schedule | Display new shipping schedule page | Display new shipping schedule page | Pass |
| TCD-0014 | Select a vessel | 1. Do not select any vessel 2. Press Create button | Display Error message | Display Error message “Please select a vessel” | Pass |
| TCD-0015 | Port of destination | 1. Do not enter any value 2. Press Create button | Display Error message | Display Error message “Please fill in Port of destination” | Pass |
| TCD-0016 | Port of arrival | 1. Do not enter any value 2. Press Create button | Display Error message | Display error message “Please fill in Port of arrival” | Pass |
| TCD-017 | Depart date | 1. Do not enter any value 2. Press Create button | Display Error message | Display error message “Please select depart date” | Pass |
| TCD-018 | Arrive date | 1. Do not enter any value 2. Press Create button | Display Error message | Display error message “Please select depart date” | Pass |

### Add new vessel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Parameter | Test Description | Expected Result | Actual Result | Status |
| TCD-0019 | Add new Vessel | 1. Click add new Vessel | Display add new vessel page | Display add new vessel page | Pass |
| TCD-0020 | Vessel Name | 1. Do not select any vessel 2. Press Create button | Display Error message | Display Error message “Please Fill in Vessel Name” | Pass |
| TCD-0021 | Description | 1. Do not enter any value 2. Press Create button | Display Error message | Display Error message “Please fill in Description” | Pass |
| TCD-0022 | Size | 1. Do not enter any value 2. Press Create button | Display Error message | Display error message “Please fill in Size” | Pass |

### Manage Agents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Parameter | Test Description | Expected Result | Actual Result | Status |
| TCD-0023 | Manage Agent | 1. Click on manage Agents | Display manage agent page | Display manage agent page | Pass |
| TCD-0024 | Approve Agent | 1. Click Approve 2. Press Create button | Display message | Display message “Agent approve” | Pass |
| TCD-0025 | Delete Agent | 1. Click Delete | Display message | Display Error message “Agent Deleted” | Pass |

### Add new customers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Parameter | Test Description | Expected Result | Actual Result | Status |
| TCD-0026 | Add new customer | 1. Click add new customer | Display add new customer page | Display add new customer page | Pass |
| TCD-0027 | Name | 1. Don’t put any value 2. Press Create button | Display error message | Display message “Please fill in customer name” | Pass |
| TCD-0028 | Address | 1. Don’t put any value 2. Press Create button | Display error message | Display message “Please fill in customer address” | Pass |

### New shipment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Parameter | Test Description | Expected Result | Actual Result | Status |
| TCD-0029 | Add new shipment | 1. Click add new shipment | Display add new shipment page | Display add new shipment page | Pass |
| TCD-0030 | Shipper | 1. Don’t select any shipper 2. Press Create button | Display error message | Display message “Please select shipper” | Pass |
| TCD-0031 | Consignee | 1. Don’t select any consignee 2. Press Create button | Display error message | Display message “Please select consignee” | Pass |
| TCD-032 | Contract number | 1. Leave it blank 2. Press create button | Create done | Create done | Pass |
| TCD-033 | Price | 1. Leave it blank 2. Press create button | Display error message | Display message “please fill in the price” | Pass |
| TCD-034 | Commodity | 1. Leave it blank 2. Press create button | Display error message | Display message “please fill in the commodity” | Pass |
| TCD-035 | Container Type | 1. Leave it blank 2. Press create button | Display error message | Display message “please fill in the container type” | Pass |
| TCD-036 | Package Amount | 1. Leave it blank 2. Press create button | Display error message | Display message “please fill in the package amount” | Pass |
| TCD-037 | Kind of package | 1. Leave it blank 2. Press create button | Display error message | Display message “please fill in the kind of package” | Pass |
| TCD-038 | Weight | 1. Leave it blank 2. Press create button | Display error message | Display message “please fill in the weight” | Pass |
| TCD-039 | Volume | 1. Leave it blank 2. Press create button | Display error message | Display message “please fill in the volume” | Pass |
| TCD-40 | Description | 1. Leave it blank 2. Press create button | Display error message | Display message “please fill in the description” | Pass |

## Performance testing

The performance of Maersk Line Container Management System can be tested by conducting the performance test of web application on the server. This can be done stimulate numbers of concurrent user making request to the web application within a period of time. The features is provided by the Microsoft Azure Portal. Each pricing tier have different number of user load to be applied and different time period to tested. For example, for basic pricing tier, the maximum number of user load is 40 concurrent users which is not sufficient to test the performance of the system while standard pricing tier allow 20,000 concurrent users to be tested for the performance of the system. The performance test will be targeted the Maersk Container Management System in South East Asia region. There will be two times of performance testing performed starting with user load of 500 concurrent users to 1000 concurrent users with each test having 250 user load increment. The figure below shows the configuration of the performance testing. It need the developer to fill up all the requirement information such as generate the load from which region, number of user load, and duration of the request form these number of user load.

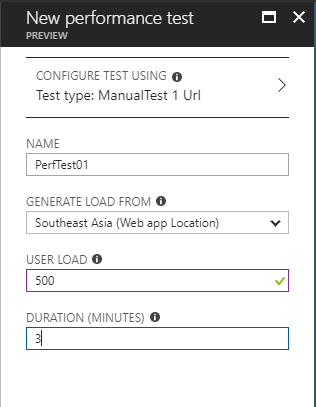


Figure 31: Configuration of performance test

After all the selection and fill in all the requirement “run” button, the performance testing is automated and it requires few minutes to complete the performance testing. The result of performance testing provides some important information in a pie chart such as number of successful request and failed request, average response time from the server, throughput of the system, CPU time and Memory working set within the period of performance testing as well as the application of scaling operation. In the first test, 500 concurrent users within 3 minutes is tested to make request to the web application. The result is 67200 successful requests and 0 failed request done to the serer which is 100% successful rate of making request to server. The average response time of the web application is 0.54 second and the request made per second is 373.33. This result is very good and acceptable. The follow figure is the result of the first performance test.

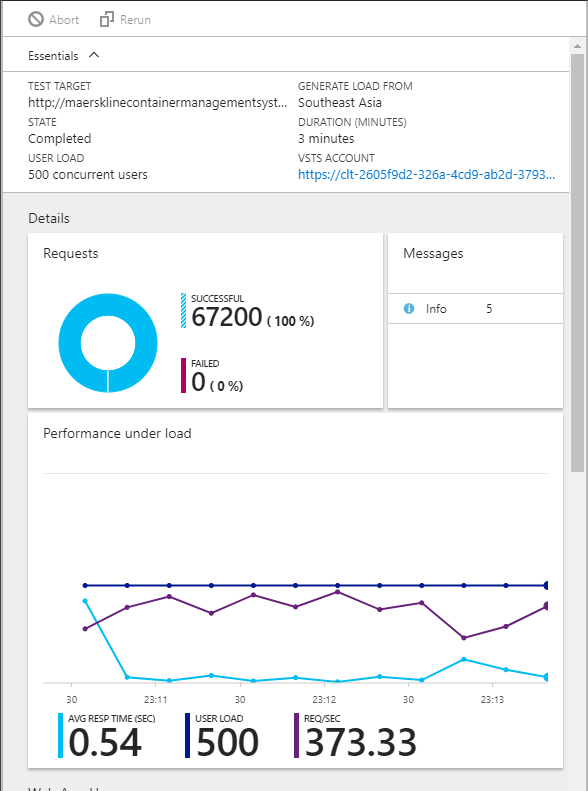


Figure 32: Result of first performance test

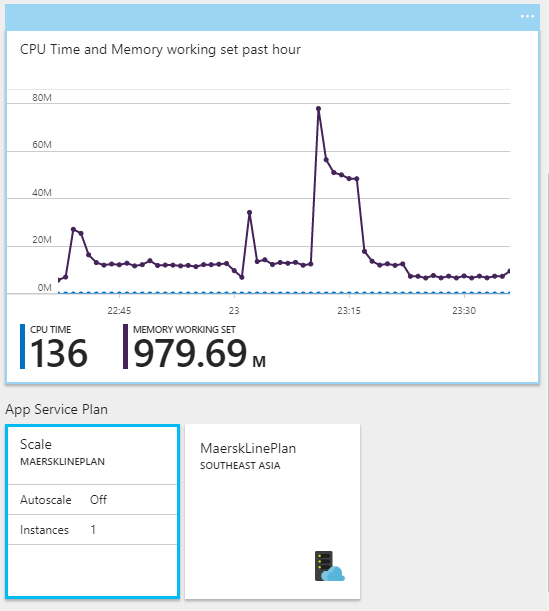


Figure 33: Result of first performance test part 2

There is two different performance test will be carried out and the result was documented in the following table.

Table 1: Performance test with different user load

|  |  |  |
| --- | --- | --- |
|  | 500 Concurrent User | 1000 Concurrent User |
| Successful Request | 67200 | 70634 |
| Failed Request | 0 | 0 |
| Average Response Time | 0.54 | 2.22 |
| Request/second | 373.33 | 392.41 |

In conclusion, the performance of the Maersk Container Management System is very good. As shown in table, there are 100% successful request made from 500 concurrent users up to 1000 concurrent users. Beside that, the average response time of the web application is good.

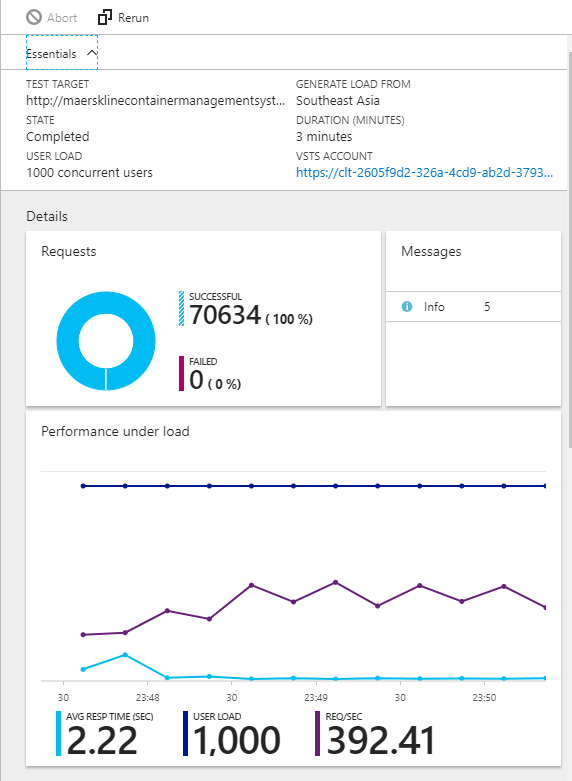


Figure 34: 1000 concurrent user test

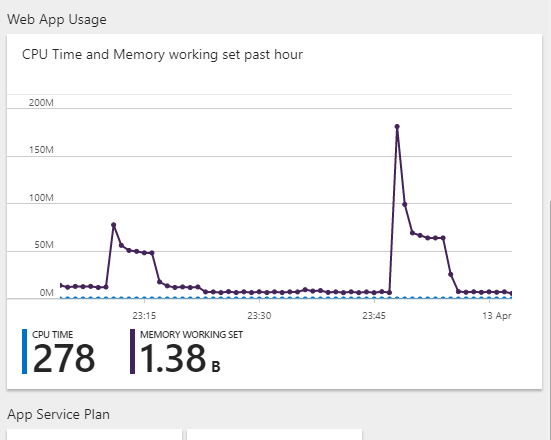


Figure 35: 1000 concurrent user test part 2

In conclusion, the S1 Standard price tier is sufficient to be applied in this web application. However, a scale up plan will be considered in the future as the web application slowly deploy to other countries or the business grow larger. Also, the development of the web application is successful and performed well because the web application could handle large amount of request at the same time without fail exclude the network issue. In conclusion, the S1 Standard price tier is sufficient to be applied in this web application. However, a scale up plan will be considered in the future as the web application slowly deploy to other countries or the business grow larger. Also, the development of the web application is successful and performed well because the web application could handle large amount of request at the same time without fail exclude the network issue.

# Azure Managed Database (Paas)

PAAS stands for Platform as Service, it is an environment established in the cloud by cloud service provider that let users to publish their application from a simple cloud-based apps to cloud-enable enterprise apps. The cloud service does provide a wide variety of service with different characteristics which let their user to purchase the desired resources and service they needed for their published system in a secure internet connection. There are few company in the market that also provide web services, company such as AliBaba, Google and Microsoft, etc.

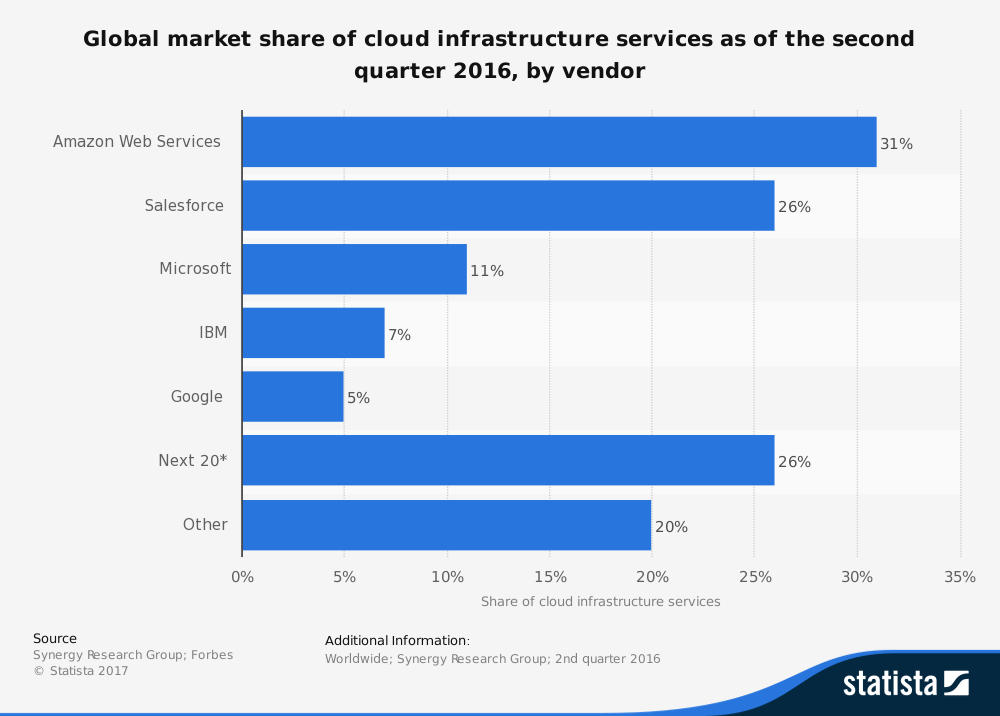


Figure 36: Cloud market share

As mention earlier, the Maersk Line Container Management System is developed based on the PAAS provided by Microsoft Azure. PAAS does not need an expert to setup and manage the server and database. First and for most, the PAAS provide different types of tools and services to boots the capability of the development team without too much of the cost to recruit people with specific skills of knowledge. Azure is very convenient which allow the developer to deploy themselves for the web application or database to the cloud with minimum of configuration.

Beside that, Azure provide PAAS that able to perform cross platform application, which makes it easier and faster to develop. Different operating system also able to use as Azure deployment can be done through the browser. Furthermore, Microsoft also provides video and tutorials about Azure to help the process of both development and deployment from early stage of the development phase(Microsoft Azure, 2018).

Apart from that, PAAS is a development platform that can be access on the internet worldwide. Which means that developer able to create server instance in any region worldwide. By doing so, it helps to improve the performance of the web application. Due to the traffic manager service can route the end-user to most least latency region to boots the performance of the system.

The purpose of relational database manage the service in Microsoft Azure. SQL Database provide different service for user, which is the SQL database, manage SQL database in elastic pool, and SQL manage Instance. Azure provide two types for the user to host the database in cloud. The two types are PAAS and IAAS (infrastructure as service) (Microsoft Azure, 2018).

Azure SQL database is the database native to cloud which is known as PAAS or database as a service (DBAAS) that is enhance for software as a service(SAAS) app development. This service do not require the administration from the user, meaning that there is someone set up and backing up and as well as operate the database(Microsoft Azure, 2018).

SQL server on the Azure Virtual Machine is Azure service which let the user to install and host in the cloud on window server virtual machine that runs on Azure which is known as IAAS(Microsoft Azure, 2018). The database administrator is require to set up and operate everything him or herself. Which means, the database administrator to take full control on the SQL server and window.

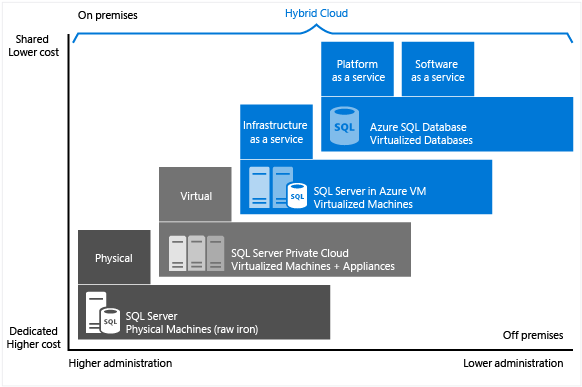


Figure 37Characteristic of level of administration over degree of cost efficiency

The figure above indicates different options to host SQL server, which are SQL Server on physical machines, SQL Server in on-premises virtualized machines, SQL Server in Azure Virtual Machines, and Azure SQL Database. Azure provides cost-effective services and it requires minimal administration on the SQL server and physical machine of the server(Microsoft Azure, 2018).

# Conclusion

In conclusion, this project was a success and fully developed with deployment on the cloud environment, the Microsoft Azure. The development of a cloud based application for Maersk Line Container management has proven to be a challenging albeit informative one. The process is not only just building but also deploying an application that fit for the cloud and utilizes all the cloud’s function that had offer and it also provide new, relevant skills which will be invaluable to the modern informational technology industry.

Throughout this project, the developer has gain much understanding about the cloud computing in its many different forms and how much can Microsoft Azure fit with the web application by publishing the Maersk Line Container management to Azure. Then the Microsoft Azure development environment has been fully explored to make full use of each and every component provided, thus the web application is more effective and efficiency. Furthermore, the developer had grow in cloud application development through the concept of realizing the practical work when designing, implementing and deploying the application on Azure. Ultimately, there is quite a good amount of knowledge gained about the architecture of efficient application for deploying on Azure. The developer believes that cloud service provide a good and strong foundation for future career path, especially the cloud application development has become a trend in this century.

# Reference

1. Microsoft Azure (2017). What is Traffic Manager. [Online] Available from: https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview. [Accessed: 11 April 2018].

2. Microsoft Azure (2018). Azure Autoscale | Microsoft Azure. [Online]. Available from: https://azure.microsoft.com/en-us/features/autoscale/. [Accessed: 11 April 2018].

3. Microsoft Azure (2018). Failover groups and active geo-replication - Azure SQL Database. [Online]. Available from: https://docs.microsoft.com/en-us/azure/sqldatabase/sql-database-geo-replication-overview. [Accessed: 11 April 2018].

4. Microsoft Azure (2018). SQL (PaaS) Database vs. SQL Server in the cloud on VMs (IaaS). [Online]. Available from: https://docs.microsoft.com/en-us/azure/sql-database/sqldatabase-paas-vs-sql-server-iaas. [Accessed: 11 April 2018].

5. Microsoft Azure (2018). What is the Azure SQL Database service?. [Online]. Available from: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-technicaloverview. [Accessed: 11 April 2018].

6. Microsoft Azure, 2014. What is PaaS?. [Online]. Available from: https://azure.microsoft.com/en-us/overview/what-is-paas/. [Accessed 12 April 2018].

7. Microsoft Azure, 2017. Autoscaling. [Online]. Available from: https://docs.microsoft.com/en-us/azure/architecture/best-practices/auto-scaling [Accessed 11 April 2018].

8. Microsoft Azure, 2017. Azure App Service Plan Overview. [Online]. Available from: https://docs.microsoft.com/en-us/azure/app-service/azure-web-sites-web-hosting-plansin-depth-overview [Accessed 11 April 2018]