

|  |  |  |
| --- | --- | --- |
| **Module Code** | : | CT071-5-3-DDAC |
| **Module Name** | : | Designing & Developing Cloud Applications |
| **Project Title** | : | Maersk Line Container Management System |
| **Student ID** | : | TP037523 |
| **Student Name** | : | Ku Ze Hao |
| **Intake Code** | : | UC3F1706SE |
| **Lecturer Name** | : | Dr. Kalai Anand A/L Ratnam |
| **Hand in Date** | : | 13 April 2018 |

# Acknowledgement

First of all, I would like to thank Dr. Kalai Anand A/L Ratnam who has taught us the course of Designing and Developing Cloud Applications. Dr. Kalai has given great tips and knowledge to allow me to better understand cloud technologies. Besides that, I would like to thank my friends that have helped me with difficulties that I have faced when developing the web application. Thanks to the help of Dr. Kalai and my friends I am able to complete this assignment successfully.

Table of Contents

[Acknowledgement II](#_Toc511347871)

[1.0 Introduction 1](#_Toc511347872)

[1.1 Project Background 1](#_Toc511347873)

[1.2 Objective 2](#_Toc511347874)

[1.3 Scope 2](#_Toc511347875)

[1.4 Summary of Major Functions 2](#_Toc511347876)

[2.0 Project Plan 3](#_Toc511347877)

[2.1 Gantt Chart 3](#_Toc511347878)

[3.0 Design 4](#_Toc511347879)

[3.1 Cloud Architecture 4](#_Toc511347880)

[3.2 Modelling 5](#_Toc511347881)

[3.2.1 Use Case Diagram 5](#_Toc511347882)

[3.2.2 Use Case Description 6](#_Toc511347883)

[3.2.3 Class Diagram 9](#_Toc511347884)

[3.2.4 Sequence Diagram 10](#_Toc511347885)

[4.0 Implementation 15](#_Toc511347886)

[4.1 Application Development 15](#_Toc511347887)

[4.2 Publishing Application & Database to Azure 18](#_Toc511347888)

[4.3 Reliability and Performances 26](#_Toc511347889)

[5.0 Test Plan & Testing Discussion 28](#_Toc511347890)

[5.1 Test Plan 28](#_Toc511347891)

[5.1.1 Admin 28](#_Toc511347892)

[5.1.2 Agent 29](#_Toc511347893)

[5.2 Performance Testing 31](#_Toc511347894)

[6.0 Implementation and Discussion on Managed Database (PaaS) 34](#_Toc511347895)

[7.0 Conclusion 36](#_Toc511347896)

[References 37](#_Toc511347897)

# 1.0 Introduction

## 1.1 Project Background

Maersk Line is the global container division and the largest operating unit of the A.P. Moller – Maersk Group, a Danish business conglomerate. Maersk Line was founded in 1928. Maersk Line currently operates in 100 countries and transports goods around the globe. Due to the volume 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 Line decided to consolidate all of its data centers and server rooms operating worldwide onto a virtualized platform. Maersk Line has decided to develop a web application which will be hosted on Microsoft Azure as Web Service for Maersk Line to handle operations like import and export.

## 1.2 Objective

The objective of this project is to develop a Maersk Line Container Management System by using Microsoft Azure as the cloud platform to solve the shipping issues and reduce supply chain costs to manage logistics efficiently.

## 1.3 Scope

Below are the scopes:

1. Import and export to gate operations.
2. Manage booking process such as schedule creation and booking confirmation.
3. Able to meet the needs of demands during peak seasons by scaling the solution.
4. Able to deliver automation of accurately allocate containers and haulier vehicles to remove human errors and reduce logistic costs.
5. Contains Failover Management for assurance and reliability.
6. Aims to improve remove human mistakes, overall profitability, increase productivity, reduce costs and optimize resources for future cargo business in high efficiency.

## 1.4 Summary of Major Functions

**Agent**

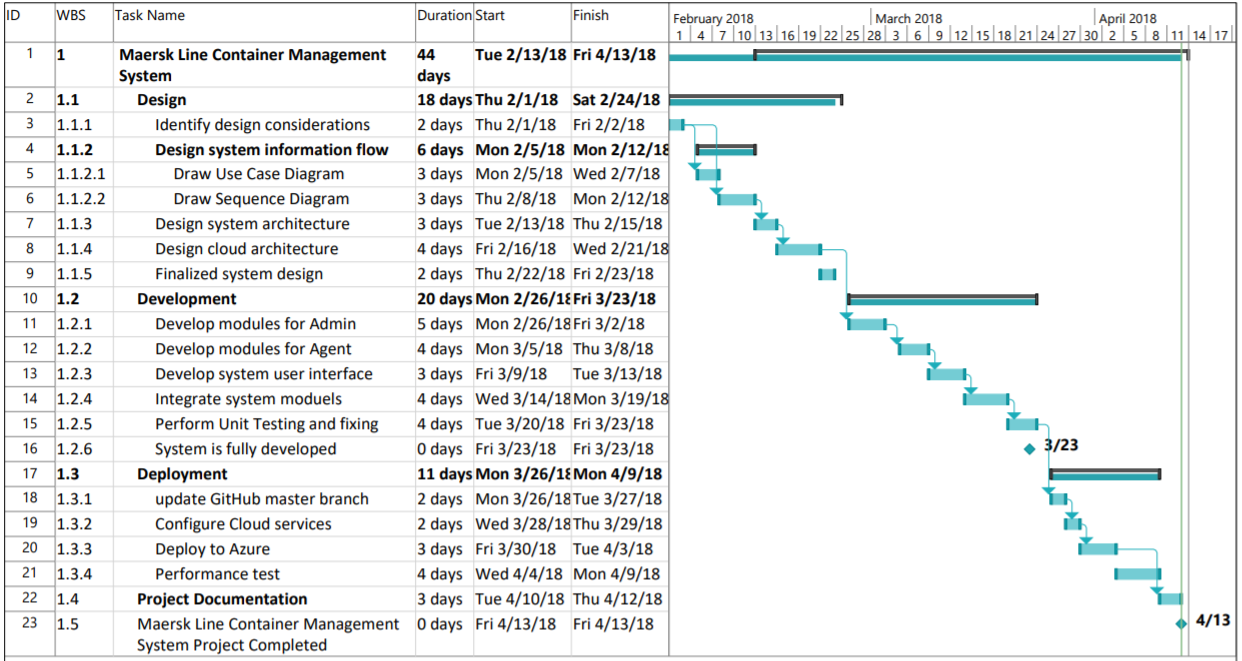
* Create Customer
* Add Item
* Assign Item
* View Item
* View Customer
* View Schedule
* View Item on Ship

**Admin**

* Register Agent
* View Schedule
* View Item on Ship
* Create Schedule

# 2.0 Project Plan

## 2.1 Gantt Chart



# 3.0 Design

## 3.1 Cloud Architecture

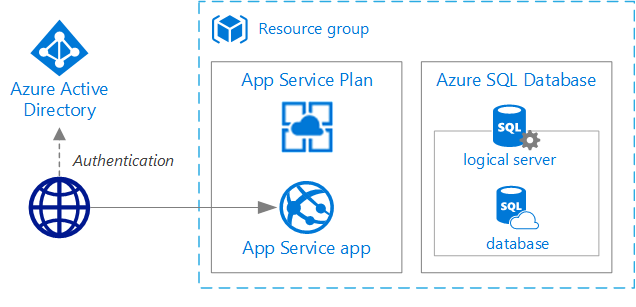


Figure 1 : Architecture Diagram

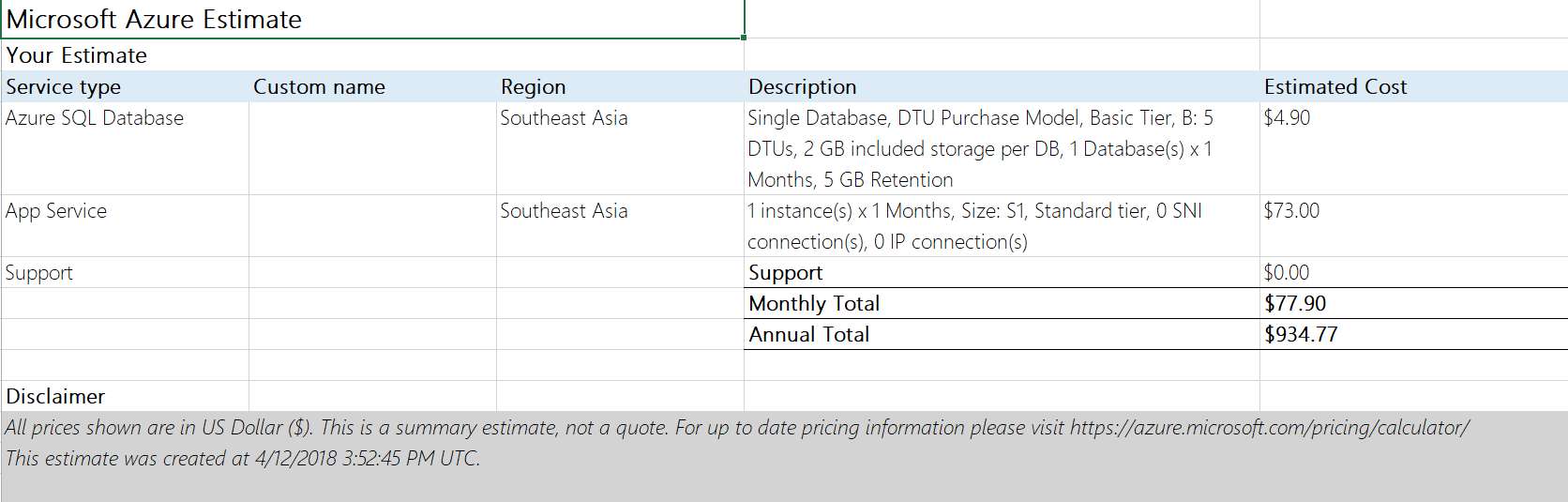


Figure 2 : Microsoft Azure Services Estimation

## 3.2 Modelling

### 3.2.1 Use Case Diagram

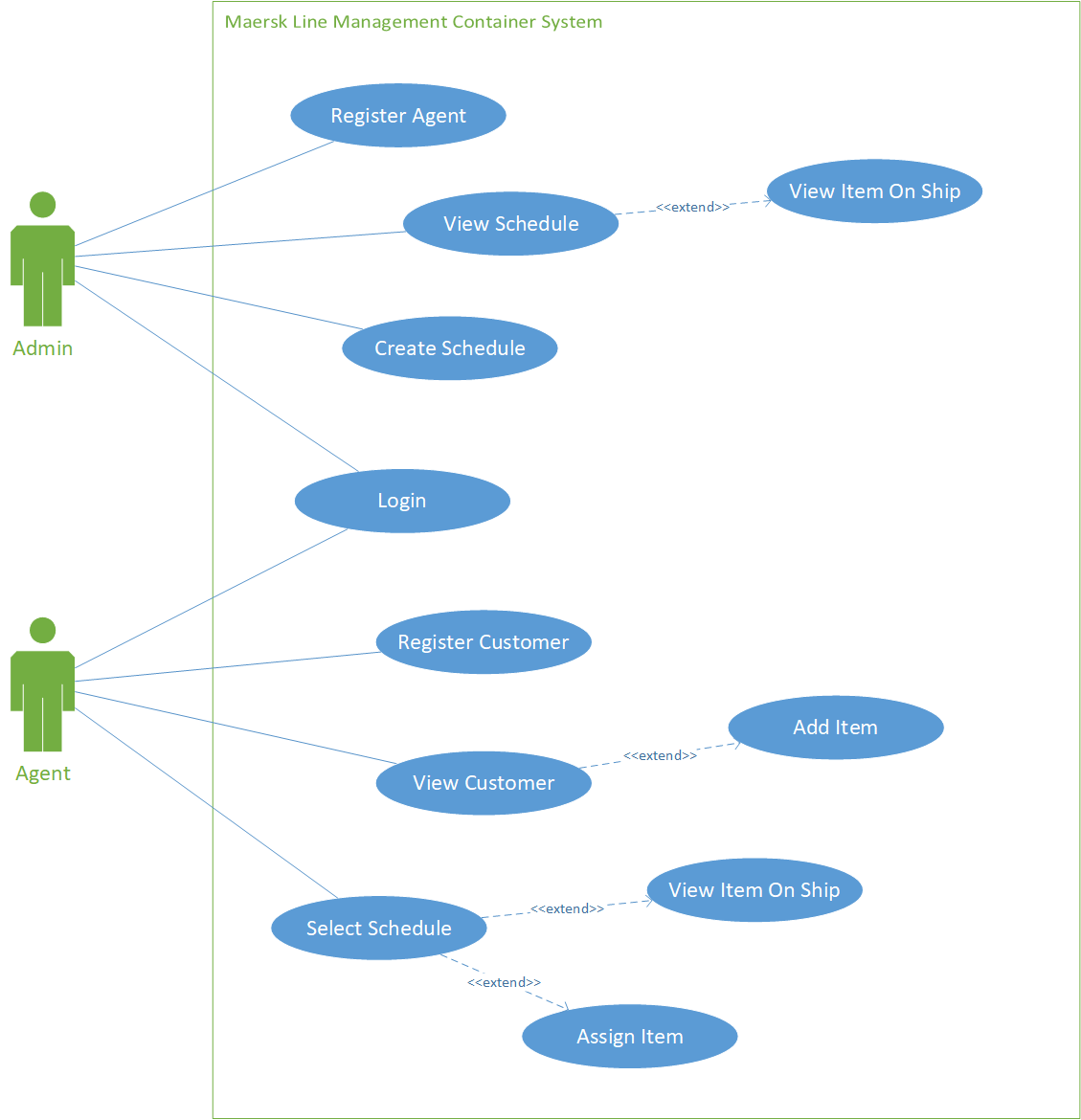


Figure 3 : Use Case Diagram

### 3.2.2 Use Case Description

**Login**

|  |  |
| --- | --- |
| ID | MLCMS-01 |
| Use Case | Login |
| Actors | Admin, Agent |
| Preconditions | None |
| Main Sequence | 1. Enter Username  2. Enter Password  3. Click Login |
| Alternative Sequence | Step 3: Username or password is wrong. Return an error message. |
| Post Condition | Loads to Admin page or agent page depending on the actor. |

**Register Agent**

|  |  |
| --- | --- |
| ID | MLCMS-02 |
| Use Case | Register Agent |
| Actors | Admin |
| Preconditions | User have to login into the system |
| Main Sequence | 1. User fill in the details  2. User confirm |
| Alternative Sequence | Step 2: Missing details. Return error Message |
| Post Condition | System updates Login table. |

**Register Customer**

|  |  |
| --- | --- |
| ID | MLCMS-03 |
| Use Case | Register Customer |
| Actors | Agent |
| Preconditions | User have to login into the system |
| Main Sequence | 1. User fill in the details  2. User confirm |
| Alternative Sequence | Step 2: Missing details. Return error Message |
| Post Condition | System updates Customer table. |

**Create Schedule**

|  |  |
| --- | --- |
| ID | MLCMS-04 |
| Use Case | Create Schedule |
| Actors | Admin |
| Preconditions | User have to login into the system |
| Main Sequence | 1. User select Create Schedule from dropdown.  2. User fills in the details  3. User confirms |
| Alternative Sequence | Step 3: Missing details. Return error Message |
| Post Condition | System updates Schedule table. |

**View Schedule**

|  |  |
| --- | --- |
| ID | MLCMS-05 |
| Use Case | View Schedule |
| Actors | Admin |
| Preconditions | User have to login into the system |
| Main Sequence | 1. User select View Schedule from dropdown. |
| Alternative Sequence | Step 1: No Schedule. Return error Message. |
| Post Condition | None |

**View Item On Ship**

|  |  |
| --- | --- |
| ID | MLCMS-06 |
| Use Case | View Item On Ship |
| Actors | Admin, Agent |
| Preconditions | User have to login into the system |
| Main Sequence | 1. User select View Schedule from dropdown(Admin) or Select Schedule(Agent).  2. User select View Item on the specified ship |
| Alternative Sequence | Step 1: No Schedule. Return error Message. |
| Post Condition | None |

**View Customer**

|  |  |
| --- | --- |
| ID | MLCMS-07 |
| Use Case | View Customer |
| Actors | Agent |
| Preconditions | User have to login into the system |
| Main Sequence | 1. User select Customer. |
| Alternative Sequence | Step 1: No customer. Return error Message. |
| Post Condition | None |

**Select Schedule**

|  |  |
| --- | --- |
| ID | MLCMS-08 |
| Use Case | Select Schedule |
| Actors | Agent |
| Preconditions | User have to login into the system |
| Main Sequence | 1. User select Select Schedule. |
| Alternative Sequence | Step 1: No Schedule. Return error Message. |
| Post Condition | None |

**Assign Item**

|  |  |
| --- | --- |
| ID | MLCMS-09 |
| Use Case | Assign Item |
| Actors | Agent |
| Preconditions | User have to login into the system |
| Main Sequence | 1. User select Add Item on Schedule.  2. User select the item that wants to be added to the ship. |
| Alternative Sequence | Step 1: Ship past Date of Departure and total item exceeded specified amount. Return error message.  Step 2: Item has already being Assigned. Return Error Message. |
| Post Condition | Update Item On Schedule and Schedule. |

**Add Item**

|  |  |
| --- | --- |
| ID | MLCMS-10 |
| Use Case | Add Item |
| Actors | Agent |
| Preconditions | User have to login into the system |
| Main Sequence | 1. User select customer.  2. User fills in Item details.  3. User confirms. |
| Alternative Sequence | Step 3: Item details not completed. Return error Message. |
| Post Condition | Update Item table. |

### 3.2.3 Class Diagram

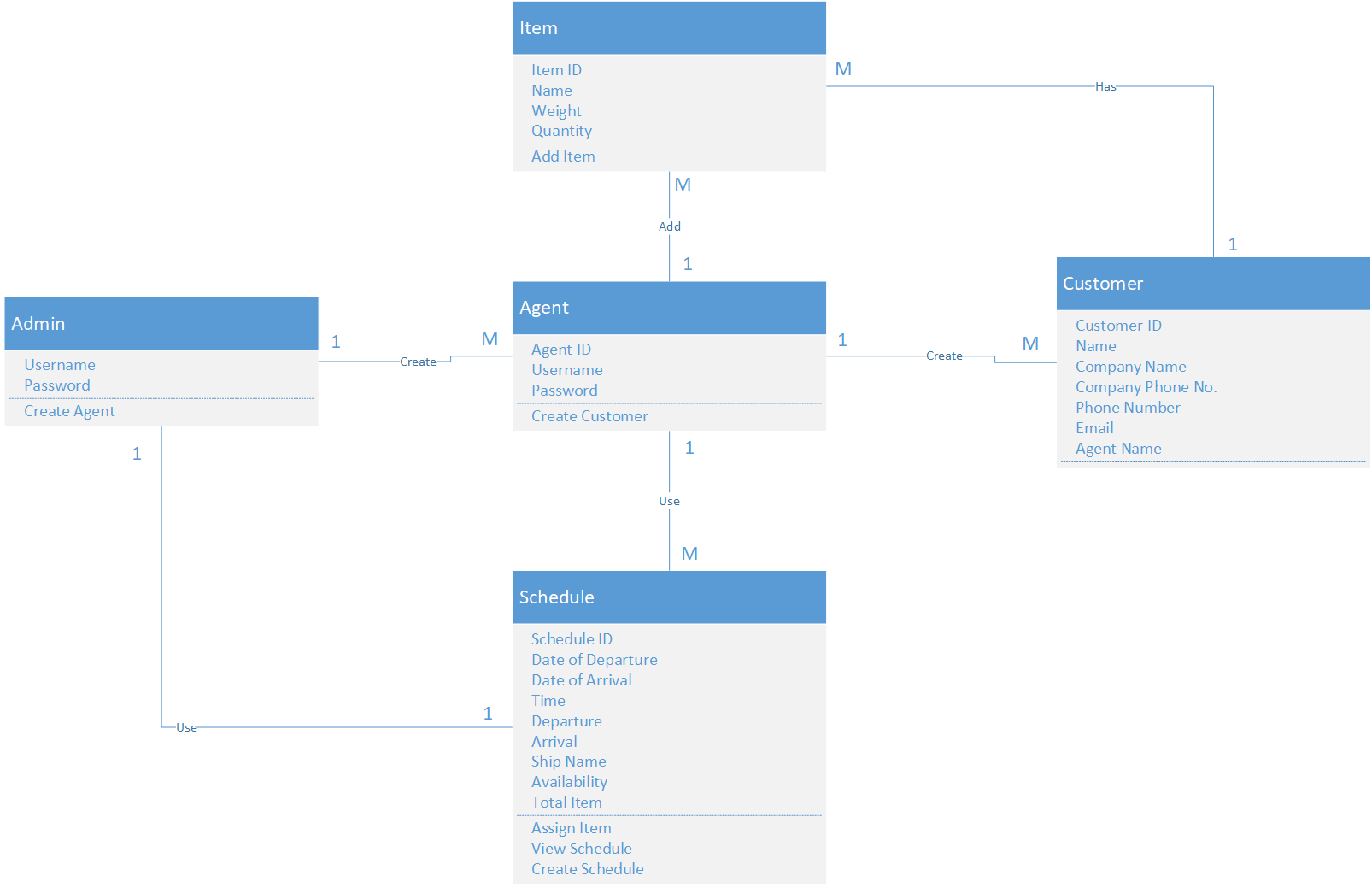


Figure 4 : Class Diagram

### 3.2.4 Sequence Diagram

**Login**

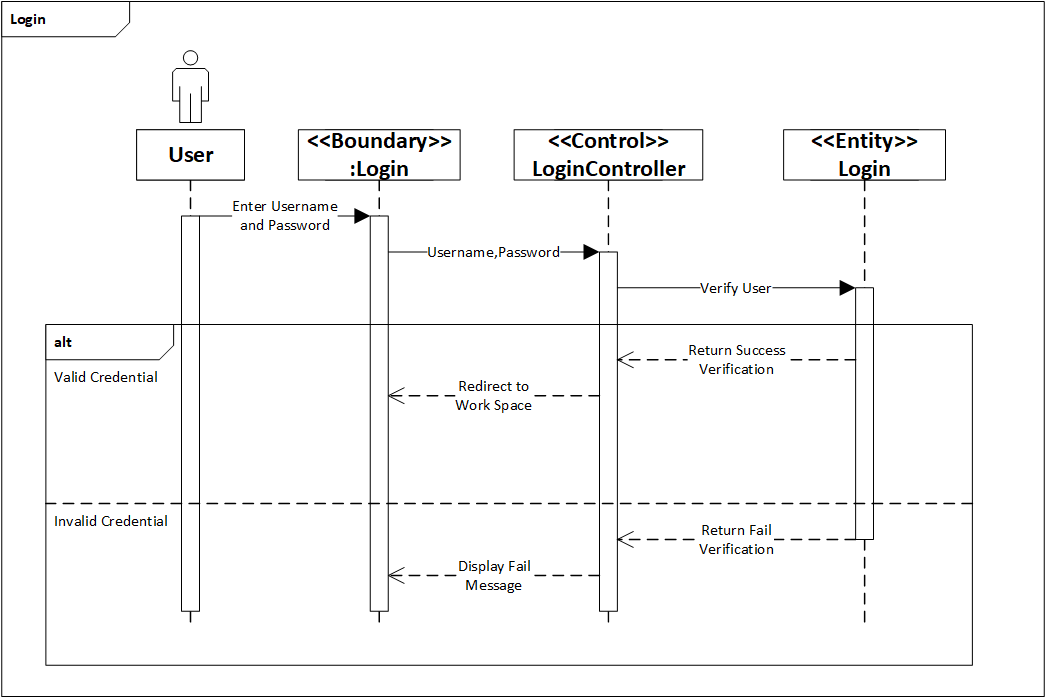
****

Figure 5 : Login Sequence Diagram

**Register Agent**

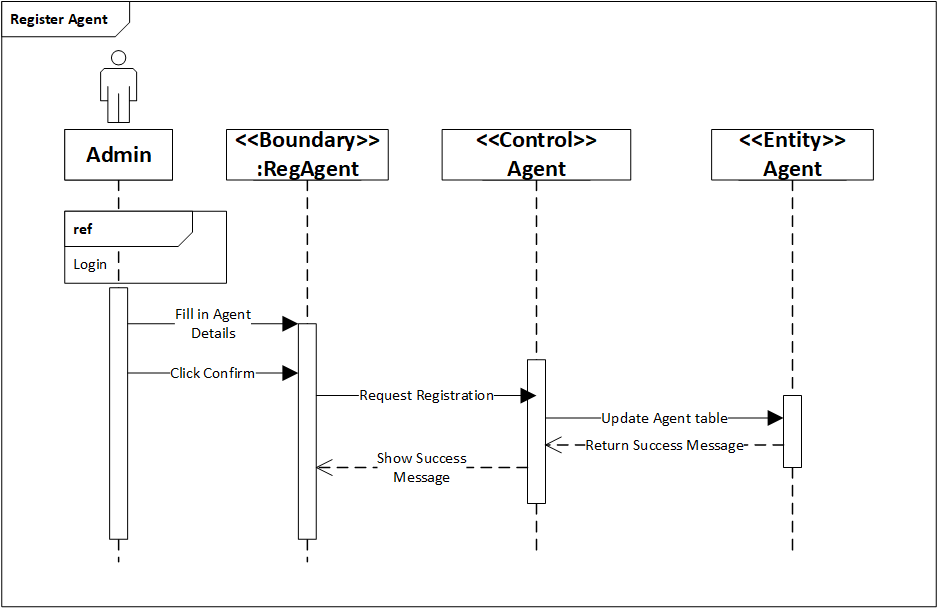
****

Figure 6 : Register Agent Sequence Diagram

**Register Customer**

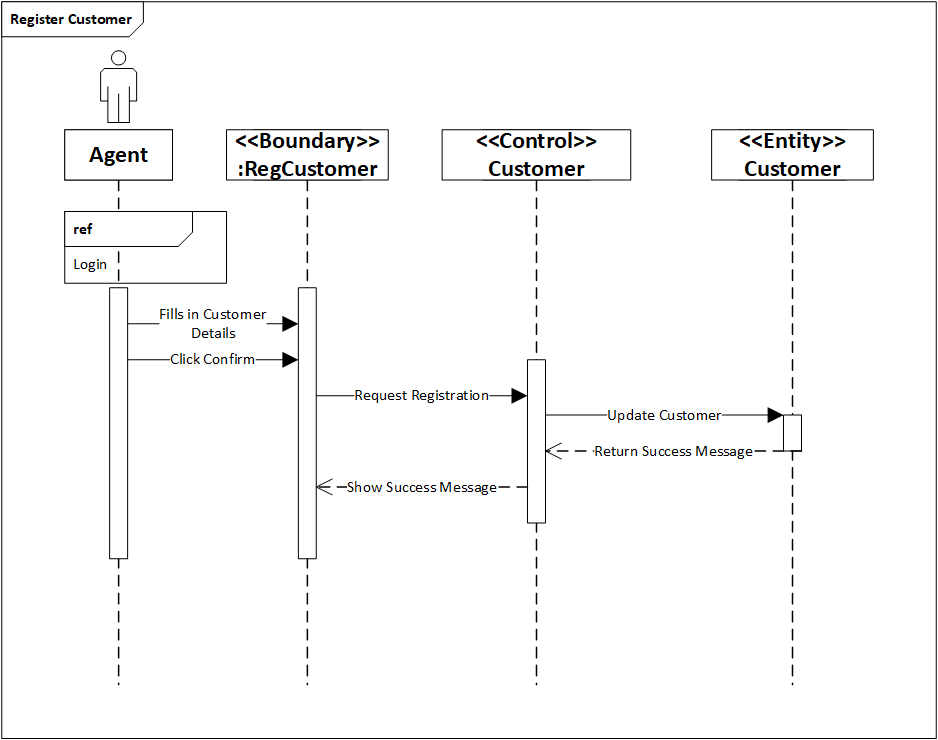
****

Figure 7 : Register Customer Sequence Diagram

**Create Schedule**

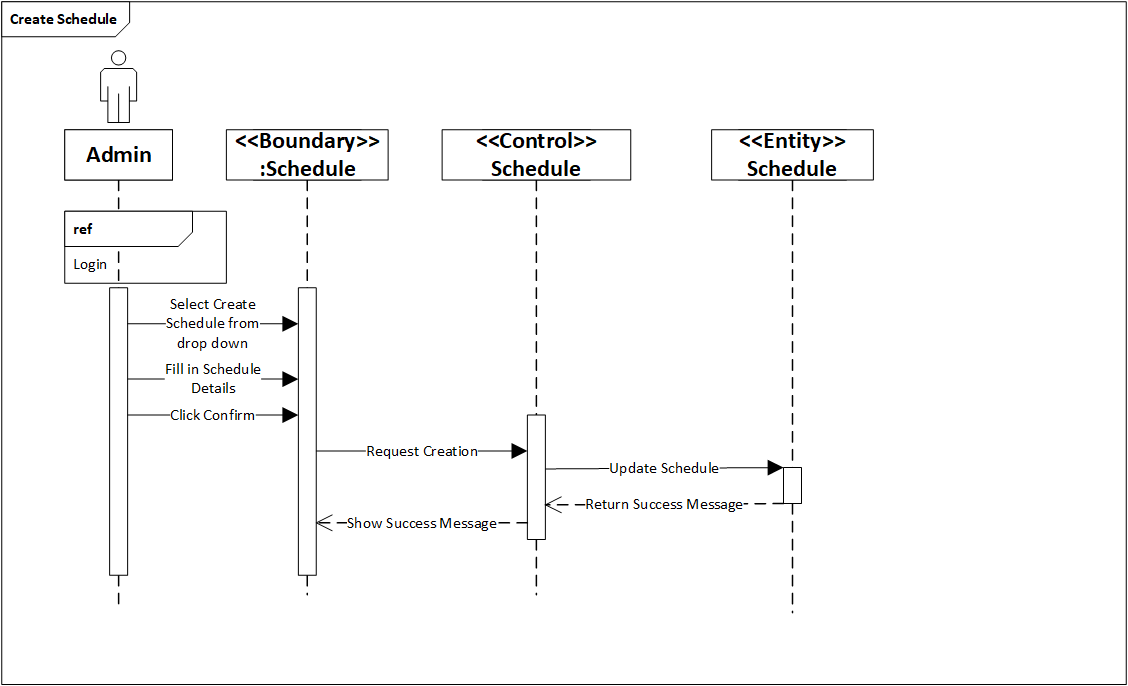
****

Figure 8 : Create Schedule Sequence Diagram

**View Schedule**

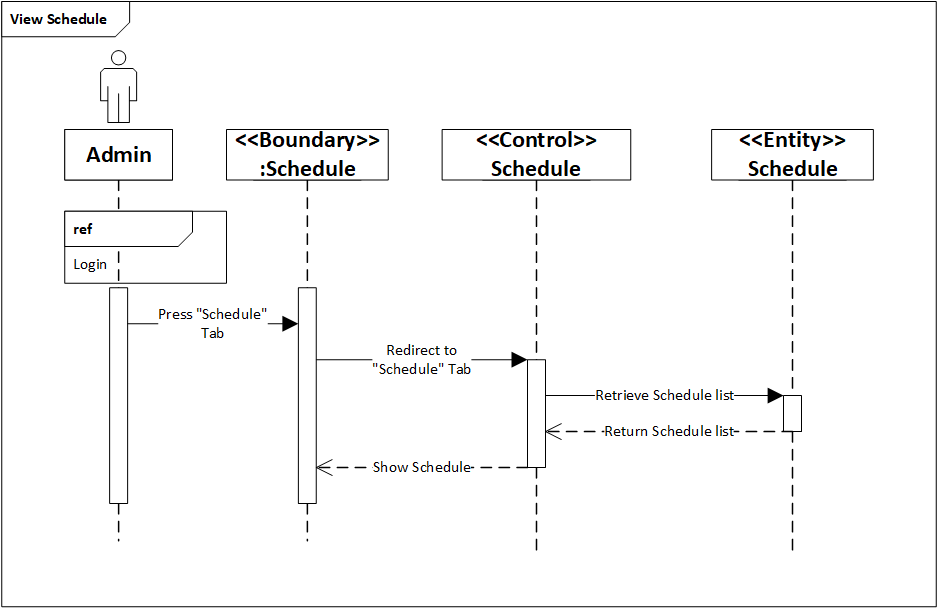
****

Figure 9 : View Schedule Sequence Diagram

**View Item On Ship**

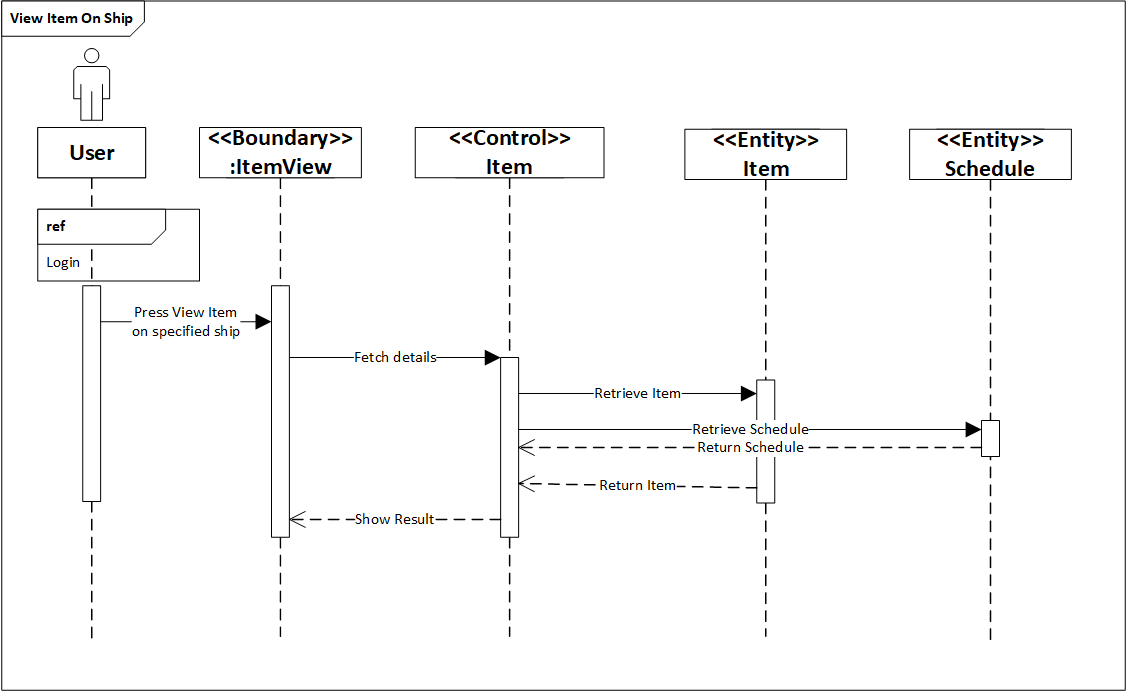
****

Figure 10 : View Item On Ship Sequence Diagram

**View Customer**

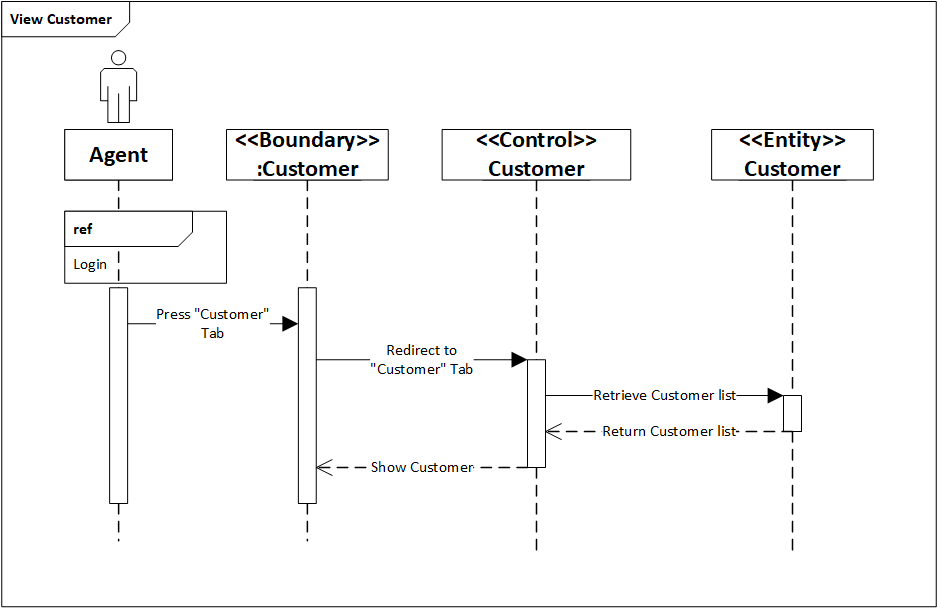
****

Figure 11 : View Customer Sequence Diagram

**Select Schedule**

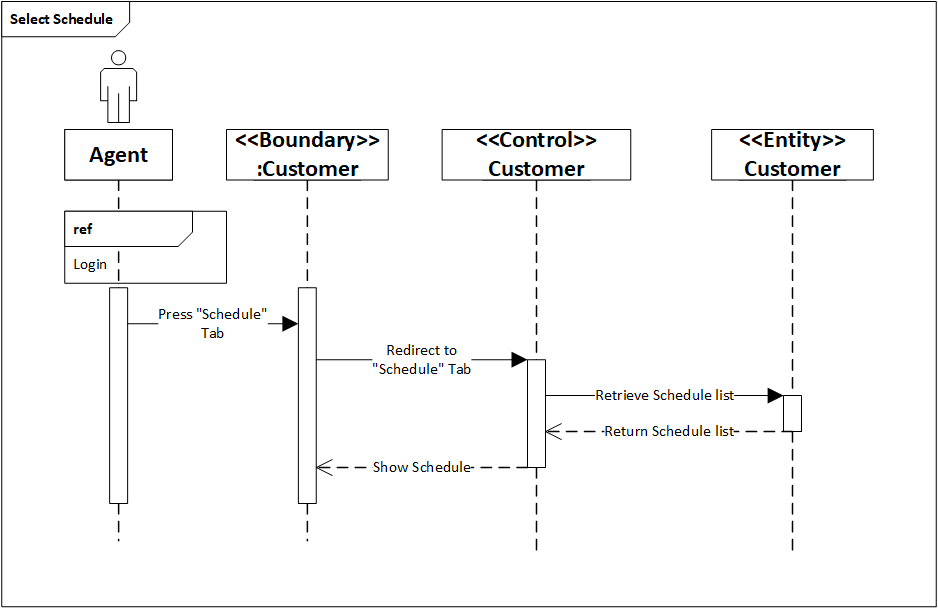
****

Figure 12 : Select Schedule Sequence Diagram

**Assign Item**

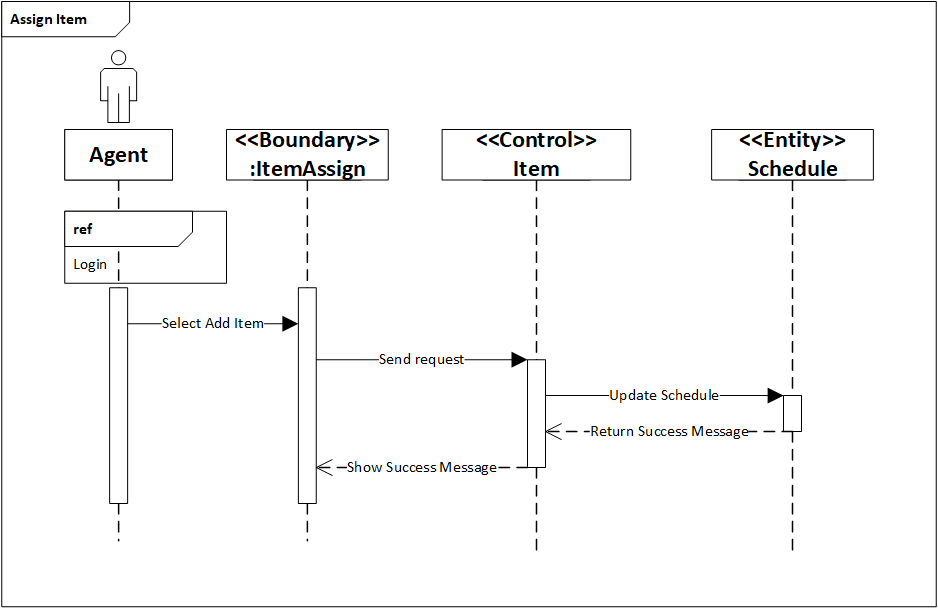
****

Figure 13 : Assign Item Sequence Diagram

**Add Item**

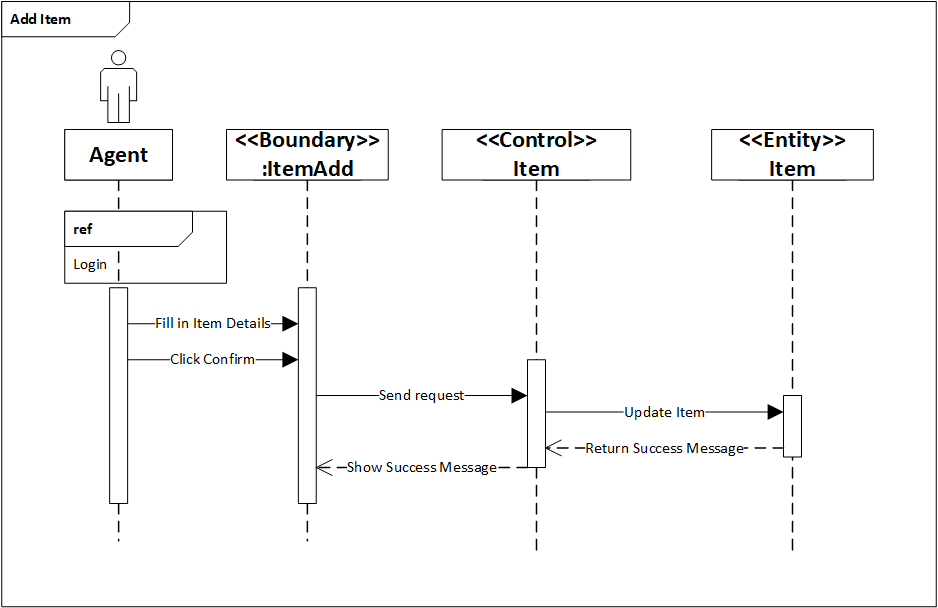
****

Figure 14 : Add Item Sequence Diagram

# 4.0 Implementation

## 4.1 Application Development

The developer has developed the Maersk Line Container Management System using Visual Studio 2015 with C# as the programming language. The database is also developed using Visual Studio 2015. ASP. NET framework is used for developing the application.

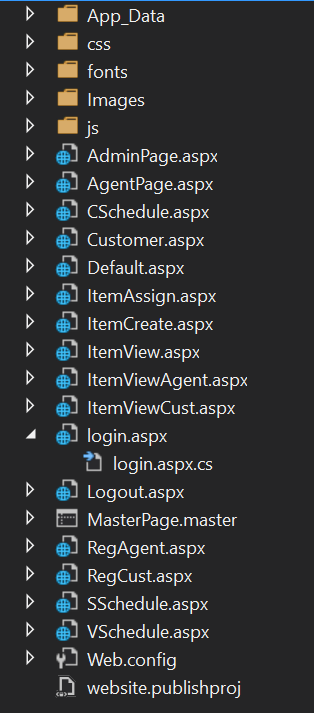


Figure 15 : System Files

Every page would follow the master page. This means that each page would have the settings of the master page which can be shown below.

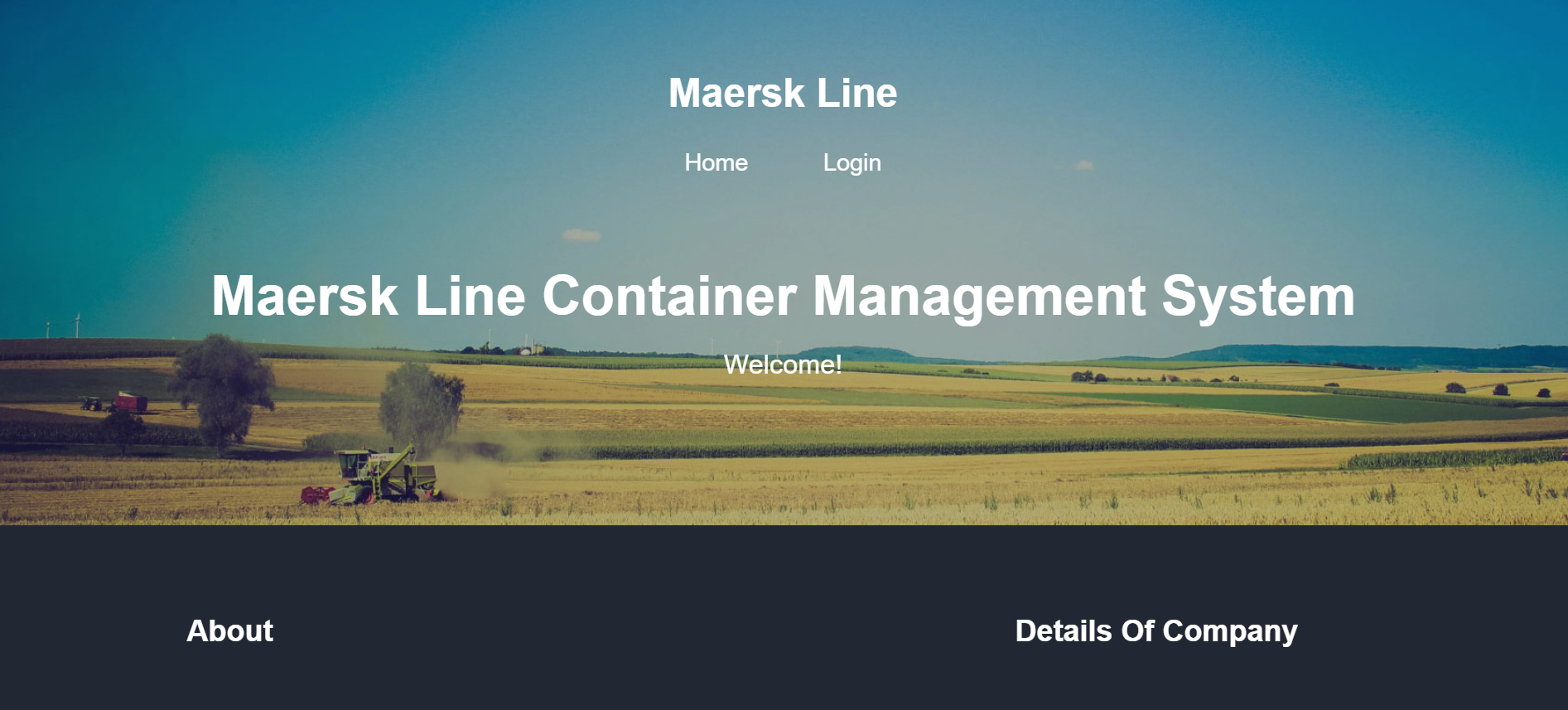


Figure 16 : Application Home Page

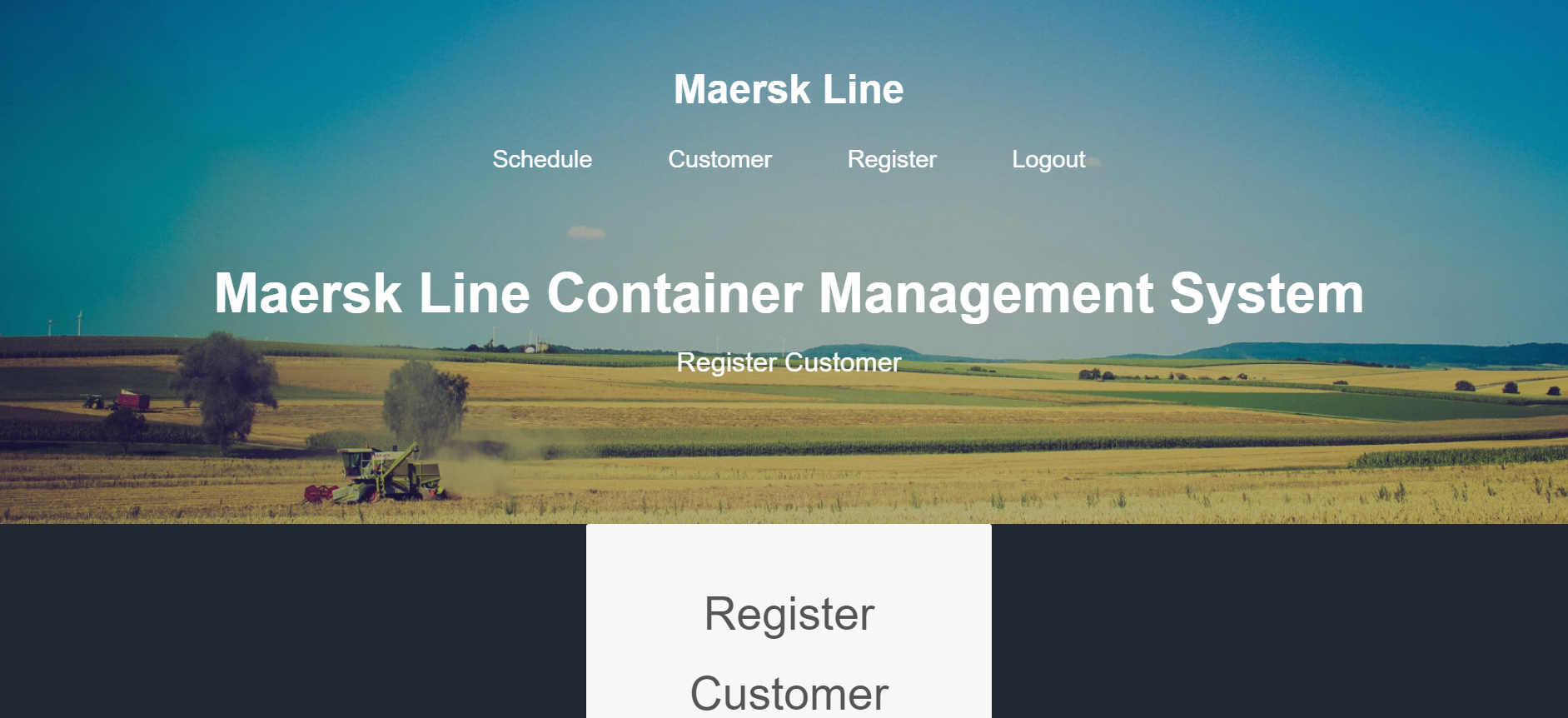


Figure 17 : Application Agent Page

Also, it shows the interface is developed using C# and html. The database would be used in local database. Inside Visual Studio 2015, the local database is shown below.

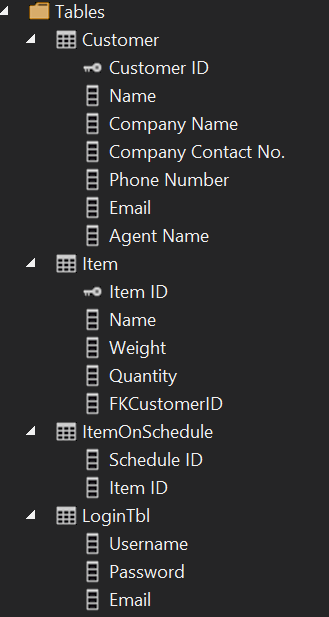
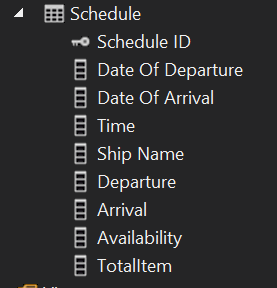


Figure 18 : Local Database

## 4.2 Publishing Application & Database to Azure

To publish the website, the developer is going to use Microsoft Azure to publish the website.

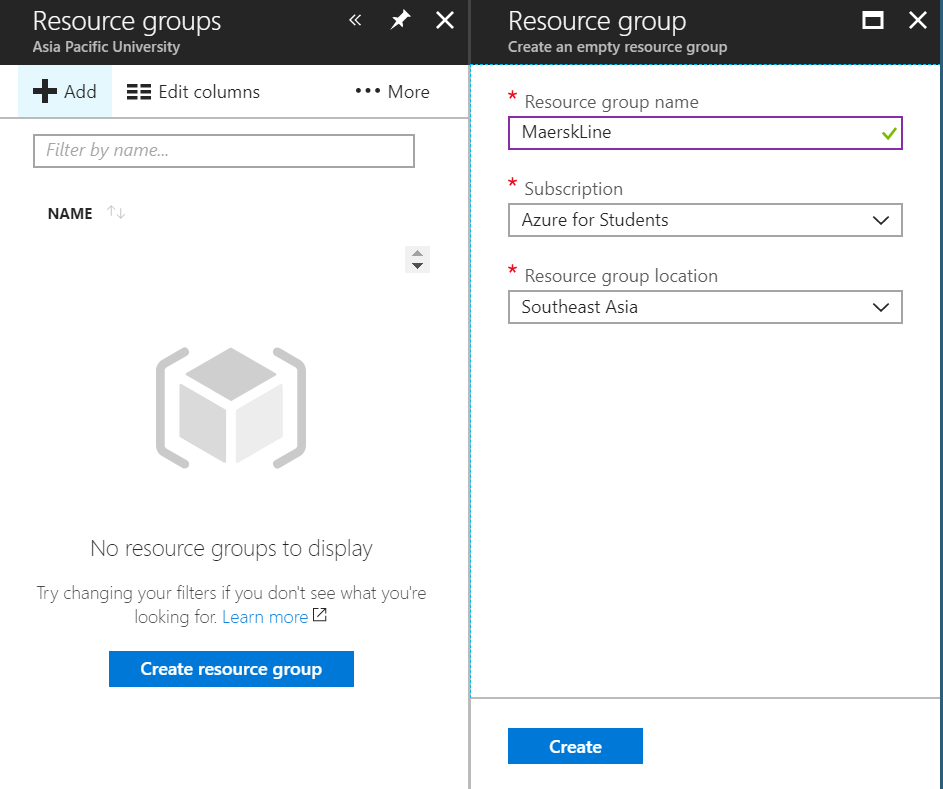


Figure 19 : Create New Resource Group

Firstly, a new resource group is created. The resource group location will be set at Southeast Asia.

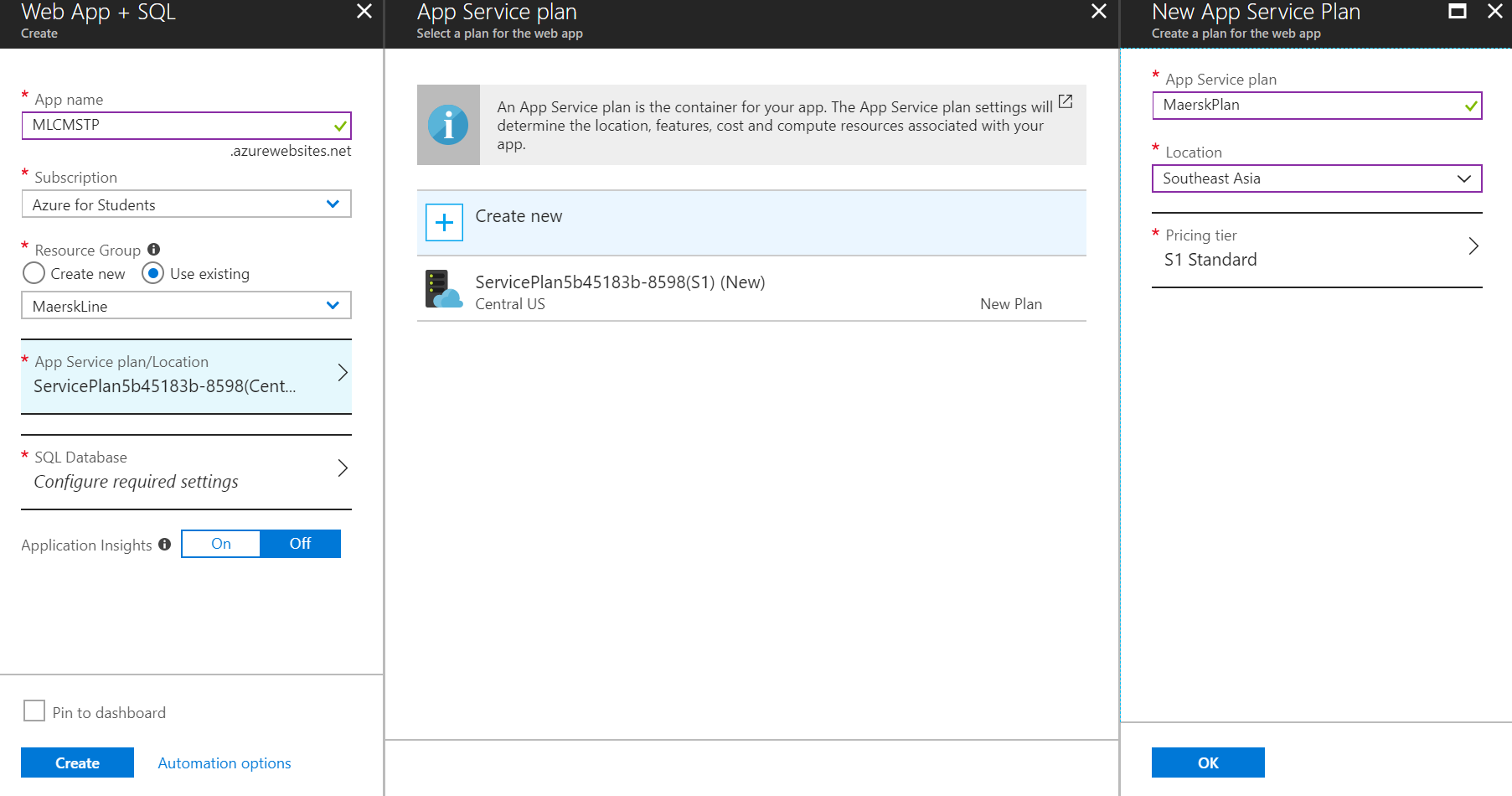


Figure 20 : Create new Web App and SQL

Inside the newly created resource group, the developer selected Create Web App + SQL. The application name is given and the resource group is double checked to be the exact same as the newly created resource group. For the App Service Plan, the developer has selected the S1 Standard and location at Southeast Asia. Figure 21 shows the current pricing standard.

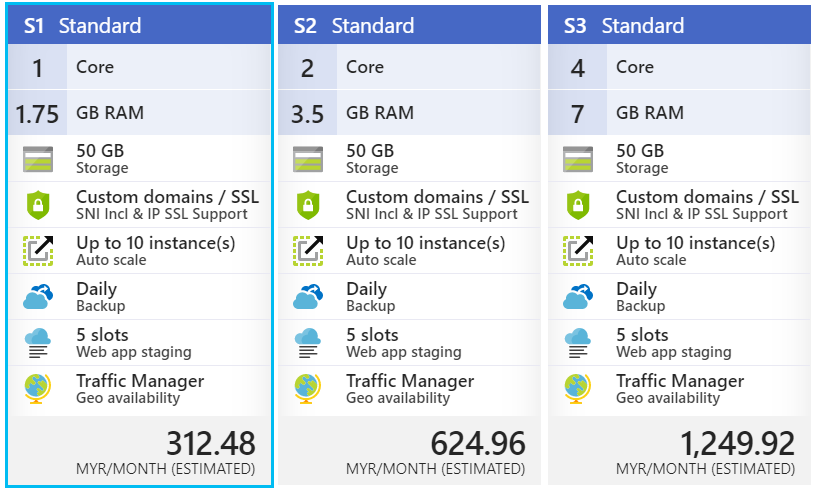


Figure 21 : App Service Plan Standard Pricing Tier

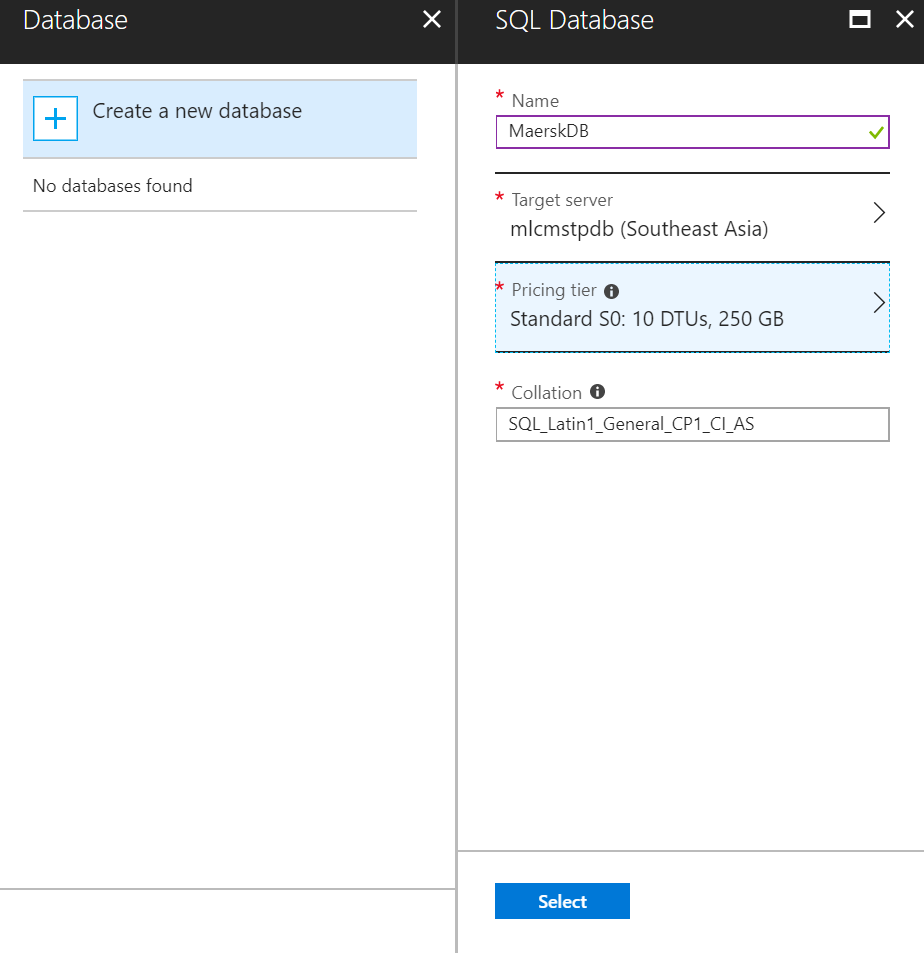


Figure 22 : Create Database

After choosing the App Service Plan, the SQL database is created. A name is given to the Azure SQL database which is Maersk DB. The server is created as shown in the figure below.

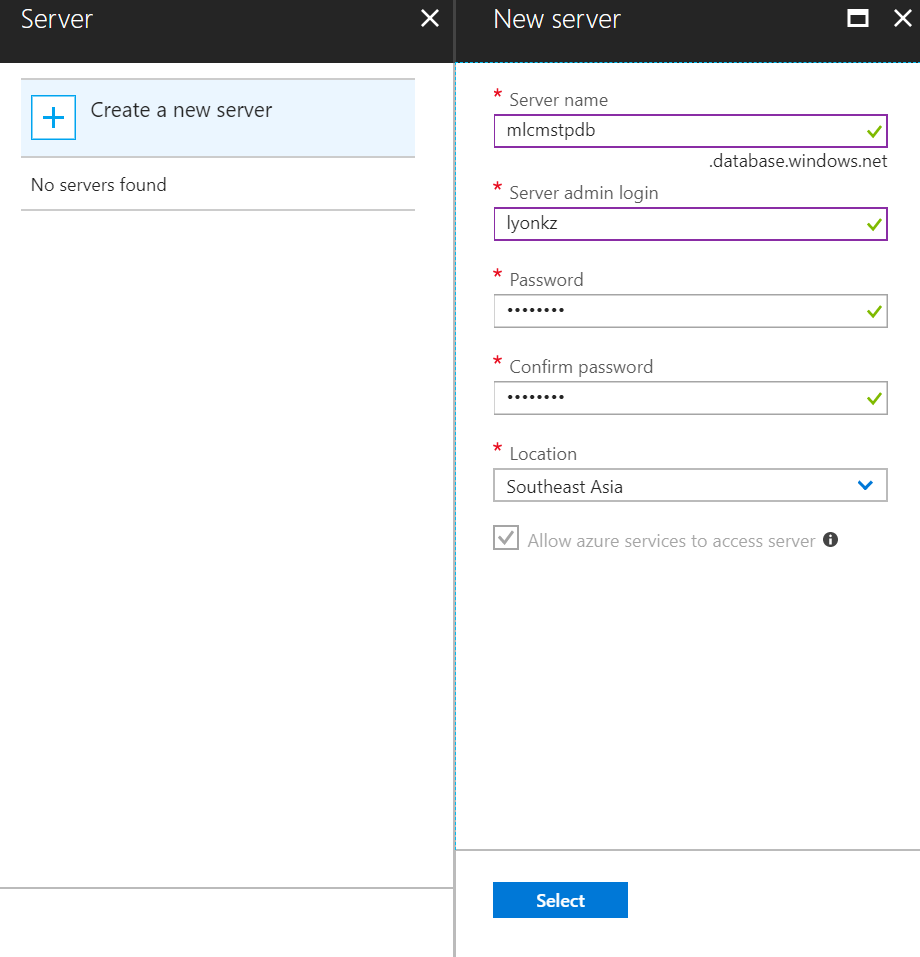


Figure 23 : Create Database Server

The pricing tier is then chosen as well as shown below.

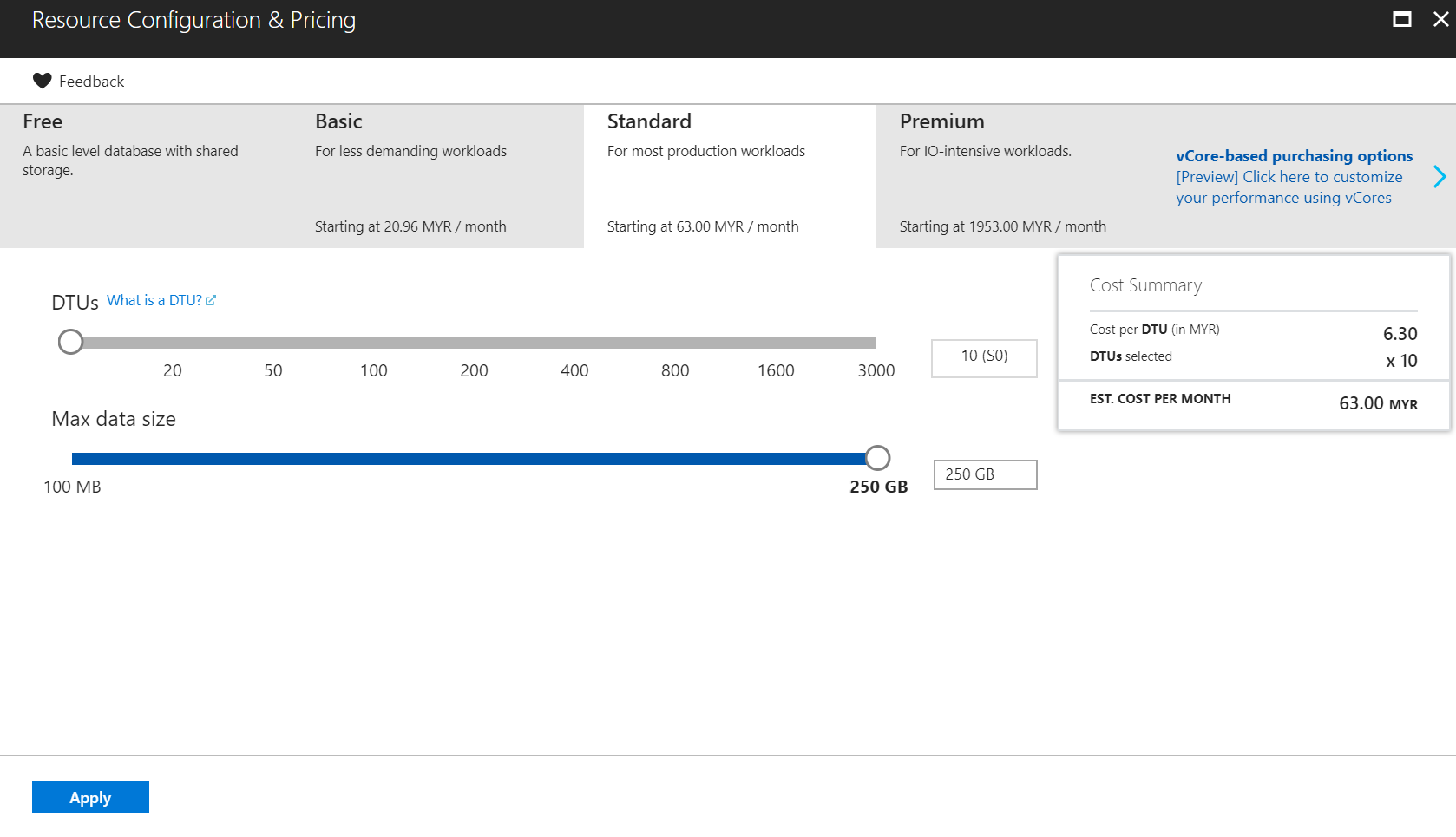


Figure 24 : Database Pricing Tier

After this, everything has been set up properly as shown below. The Web App + SQL is ready to be created.

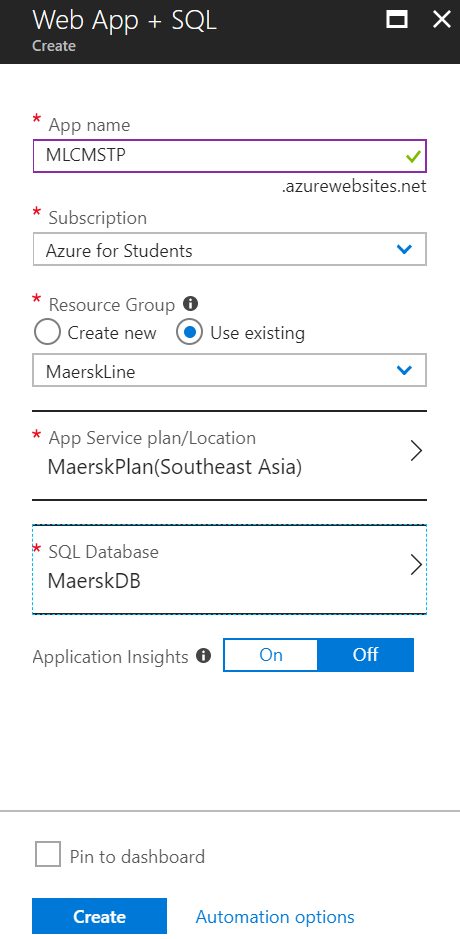


Figure 25 : Completed Configurations for Web App +SQL

Before continuing to the Visual Studio to publish the website, the firewall for the database there require a client IP address to be added. As shown below, a Client IP address is added to allow access from the client to the Azure database.

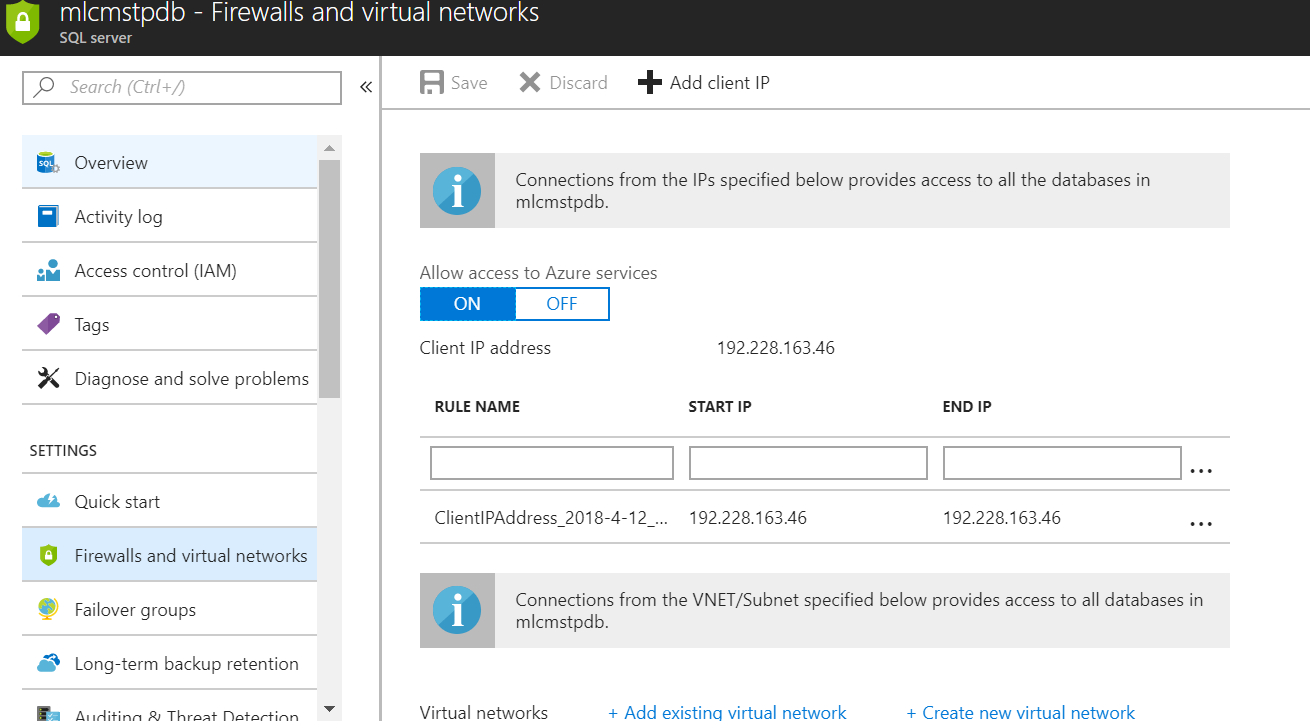


Figure 26 : Add Client IP

At the Visual Studio side, the project is published. The developer has to first create a profile by linking to Microsoft Azure and its existing Web Apps.

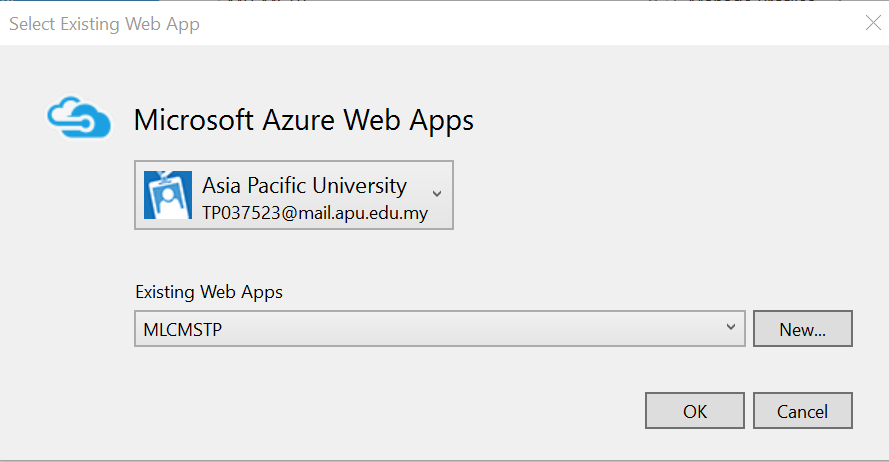


Figure 27 : Create a Profile

After creating the profile, the connection is validated.

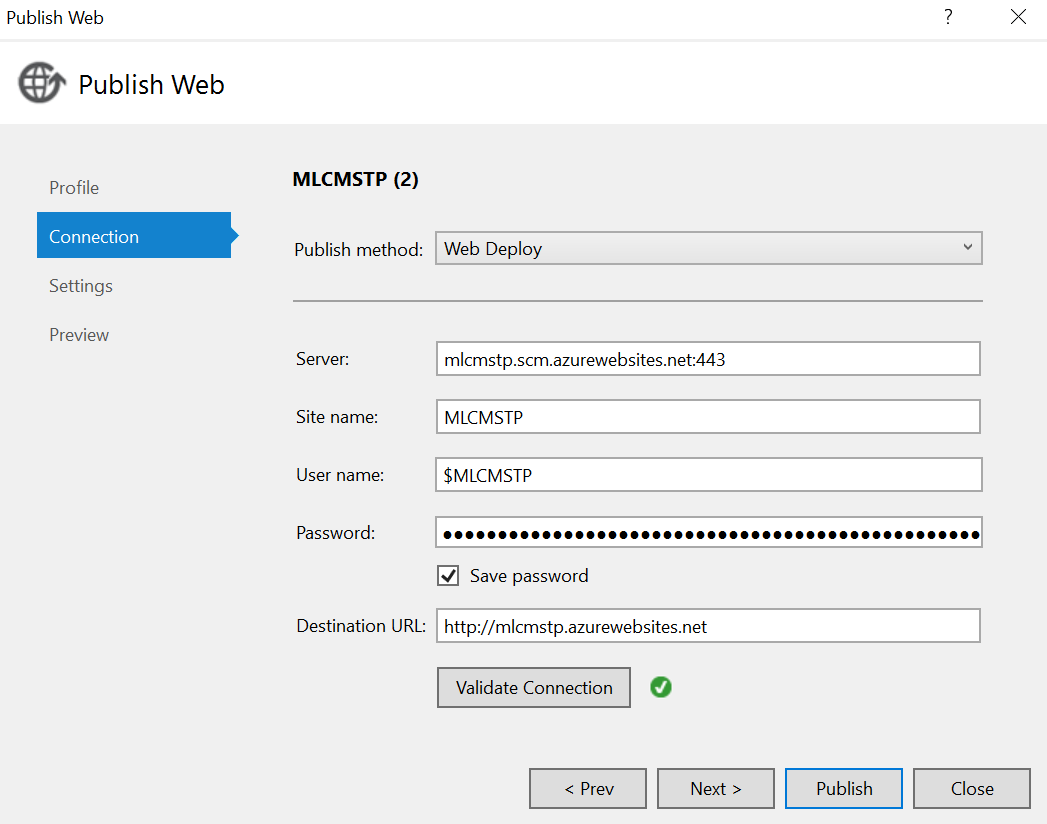


Figure 28 : Validate Connection

After that the database from Visual Studio is configured as shown below to allow for the database to be published to Microsoft Azure. After that, the website is published to Microsoft Azure.

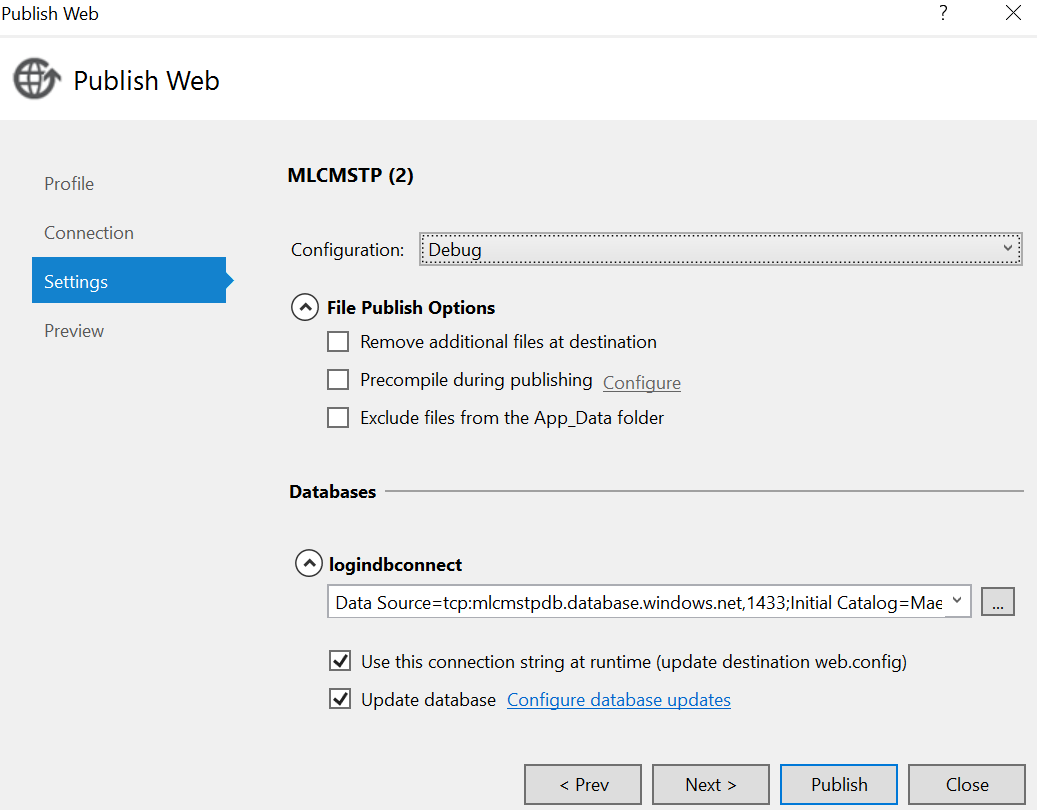


Figure 29 : Database Configuration

## 4.3 Reliability and Performances

Currently the Maersk Line Container Management System is running on one region only, if it were to expand in the future. The reliability and performances of the system need to be considered. Microsoft Azure provides Traffic Manager to allow user to control the distribution of user traffic for service endpoints in different regions. The steps on how the client request a page is shown below. (Docs.microsoft.com, 2018)

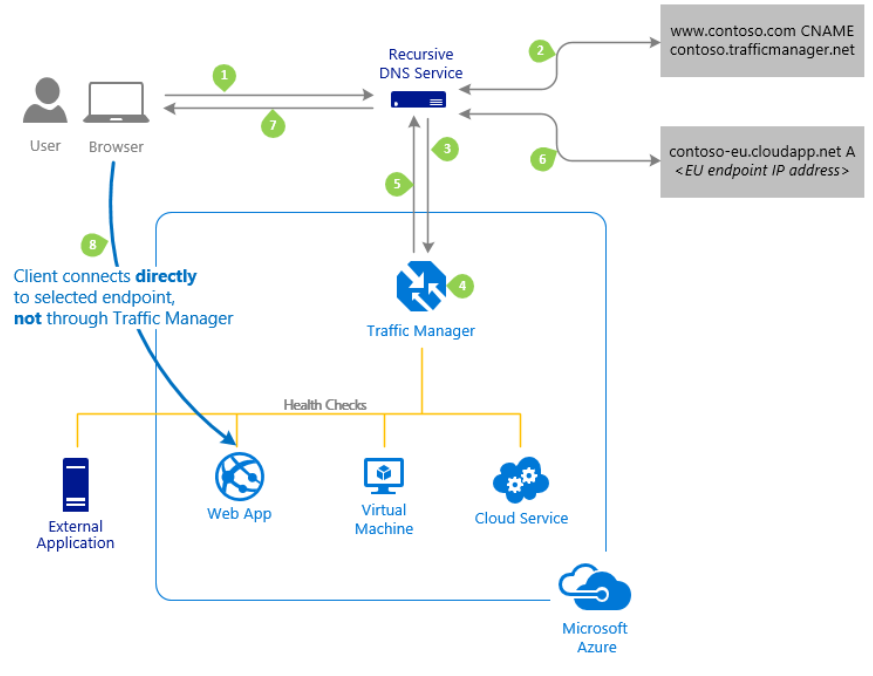


Figure 30 : Steps Client Request a Page

The price for traffic manager is based on the number of DNS queries received, with a discount for service receiving more than 1 billion monthly queries. More details of the pricing is shown below.

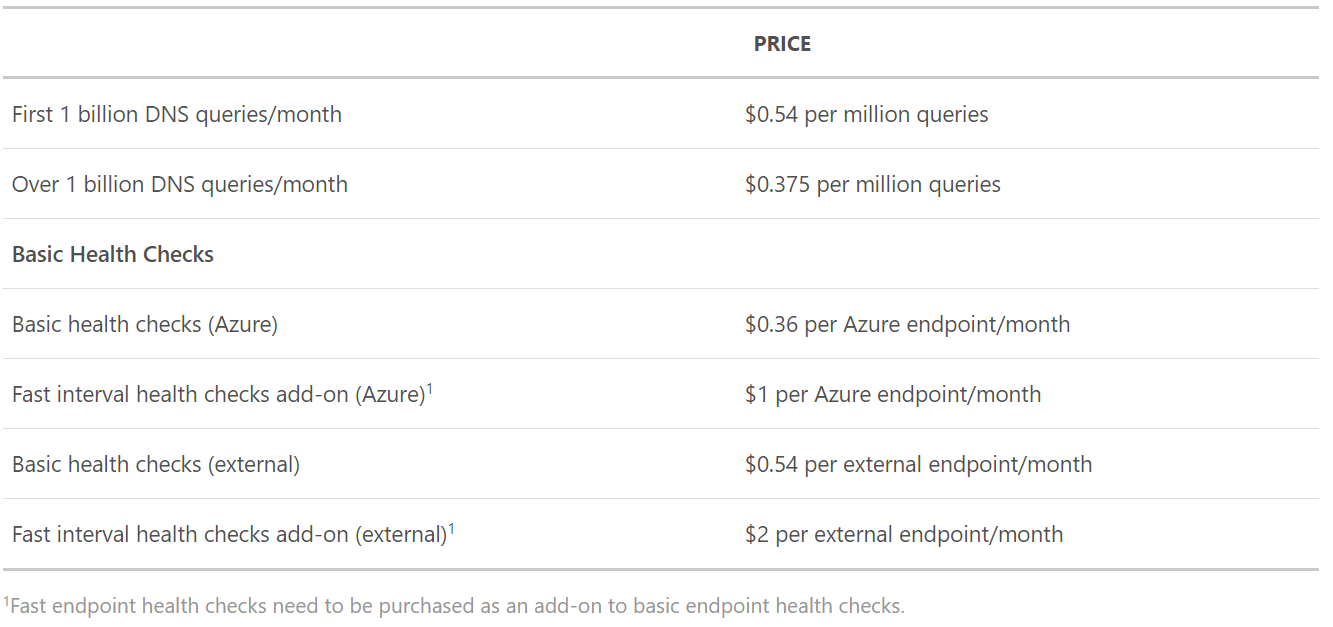


Figure 31 : Traffic Manager Pricing

# 5.0 Test Plan & Testing Discussion

## 5.1 Test Plan

### 5.1.1 Admin

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Unit | Test Case | Steps | Expected Result | Actual Result |
| TAD001 | Login | Login Success as Admin | 1. Enter Username  2. Enter Password  3. Click Login Button | Show login success result and redirect user to Admin Interface. | Show login success result and redirect user to Admin Interface. |
| TAD002 | Login User Credential Invalid | 1. Enter random Username  2. Enter random Password  3. Click Login Button | Show Error Message. | Show Error Message. |
| TAD003 | Login User Credential missing | 1. Click Login Button | Show Error Message. | Show Error Message. |
| TAD004 | Register Agent | Register Agent | 1. Select ‘Register’ tab  2. Fill in Agent details  3. Click Register button | Show agent registration success and add agent to database. | Show agent registration success and add agent to database. |
| TAD005 | Details not filled in completely | 1. Fill in Agent name  2. Click Register button | Show Error Message. | Show Error Message. |
| TAD006 | View Schedule | View Schedule | 1. Select ‘View Schedule’ from ‘Schedule’ drop down | Display Schedule. | Display Schedule. |
| TAD007 | View Schedule when no schedule exists | 1. Select ‘View Schedule’ from ‘Schedule’ drop down | Show ‘No Schedule Exists’ message. | Show ‘No Schedule Exists’ message. |
| TAD008 | Create Schedule | Create Schedule | 1. Select ‘Create Schedule’ from ‘Schedule’ drop down  2. Fill in ship details  3. Click Create button | Show schedule creation success and add schedule to database. | Show schedule creation success and add schedule to database. |
| TAD009 | Create Schedule details not filled in completely | 1. Fill in part of the details.  2. Click Create button | Show Error Message. | Show Error Message. |
| TAD010 | View Item on Ship | View Item on Ship | 1. Select ‘Schedule’ tab  2. Select ‘View Item’ on specified schedule. | Item on ship interface displayed. | Item on ship interface displayed. |
| TAD011 | No Item is on Ship | 1. Select ‘Schedule’ tab  2. Select ‘View Item’ on specified schedule. | Show ‘No Item Assigned’ message. | Show ‘No Item Assigned’ message. |

### 5.1.2 Agent

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Unit | Test Case | Steps | Expected Result | Actual Result |
| TAG001 | Login | Login Success as Agent | 1. Enter Username  2. Enter Password  3. Click Login Button | Show login success result and redirect user to Agent Interface. | Show login success result and redirect user to Agent Interface. |
| TAG002 | Login User Credential Invalid | 1. Enter random Username  2. Enter random Password  3. Click Login Button | Show Error Message. | Show Error Message. |
| TAG003 | Login User Credential missing | 1. Click Login Button | Show Error Message. | Show Error Message. |
| TAG004 | Register Customer | Register Customer | 1. Select ‘Register’ tab  2. Fill in Customer details  3. Click Register button | Show customer registration success and add customer to database. | Show customer registration success and add customer to database. |
| TAG004 | Details not filled in completely | 1. Fill in part of Customer Details  2. Click Register button | Show Error Message. | Show Error Message. |
| TAG005 | View Schedule | View Schedule | 1. Select ‘Schedule’ tab | Display Schedule. | Display Schedule. |
| TAG006 | View Schedule when no schedule exists | 1. Select ‘Schedule’ tab | Show ‘No Schedule Exists’ message. | Show ‘No Schedule Exists’ message. |
| TAG007 | View Customer | View Customer | 1. Select ‘Customer’ tab | Display Customer. | Display Customer. |
| TAG008 | View Customer when no customer exists | 1. Select ‘Customer’ tab | Show ‘No Customer Exists’ message. | Show ‘No Customer Exists’ message. |
| TAG009 | Create Item | Create Item | 1. Select ‘Customer’ tab  2. Select ‘Add Item’ on specified customer.  3. Fill in item details.  4. Click Add button. | Show item creation success and add item to database. | Show item creation success and add item to database. |
| TAG010 | Details not filled in completely | 1. Fill in part of Item Details  2. Click Add button | Show Error Message. | Show Error Message. |
| TAG011 | Assign Item | Assign Item | 1. Select ‘Schedule’ tab  2. Select ‘Assign Item’ on specified schedule  4. Select ‘Assign’ to add item to schedule. | Item successfully added to schedule. Schedule is updated based on total amount on the ship. | Item successfully added to schedule. Schedule is updated based on total amount on the ship. |
| TAG012 | No Item exists | 1. Select ‘Schedule’ tab  2. Select ‘Assign Item’ on specified schedule | Show ‘No Item Exists’ Message. | Show ‘No Item Exists’ Message. |
| TAG013 | Ship Not Available | 1. Select ‘Schedule’ tab  2. Select ‘Assign Item’ on specified schedule | Show ‘Ship is Not Available’ Message. | Show ‘Ship is Not Available’ Message. |
| TAG014 | View Item on Ship | View Item on Ship | 1. Select ‘Schedule’ tab  2. Select ‘View Item’ on specified schedule. | Item on ship interface displayed. | Item on ship interface displayed. |
| TAG015 | No Item is on Ship | 1. Select ‘Schedule’ tab  2. Select ‘View Item’ on specified schedule. | Show ‘No Item Assigned’ message. | Show ‘No Item Assigned’ message. |

## 5.2 Performance Testing

The Maersk Line Container Management System that is located in Southeast Asia will be tested with App Service Plan of Standard 1 (S1).

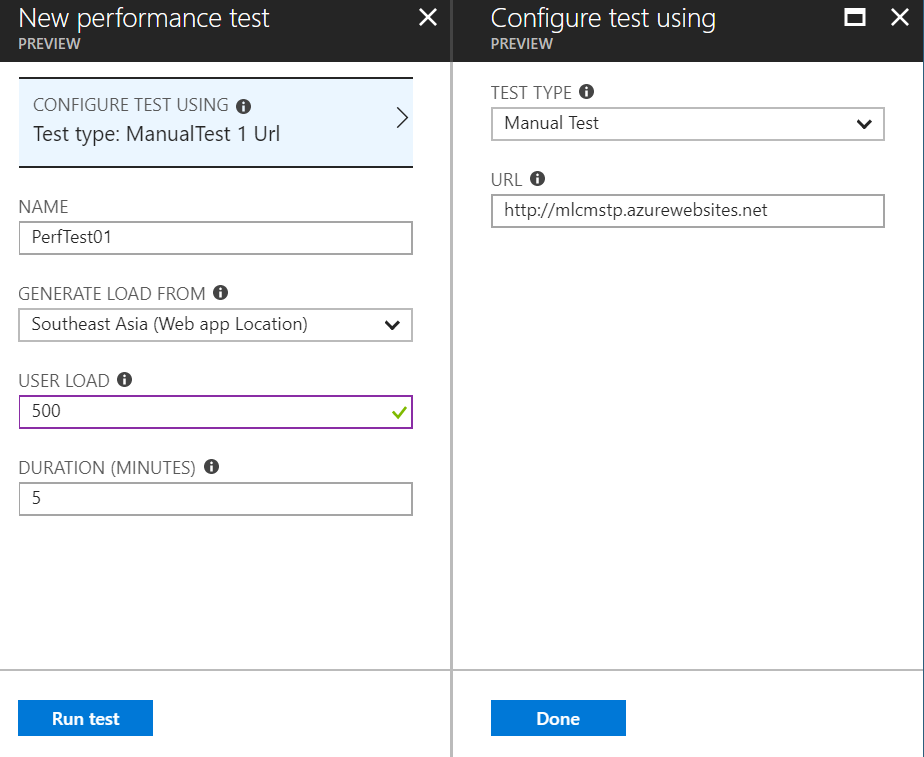


Figure 32 : Add Performance Test

The developer is planning to test a user load of 500 in 5 minutes. The test is run for about 15 minutes. The result of the test are shown below.

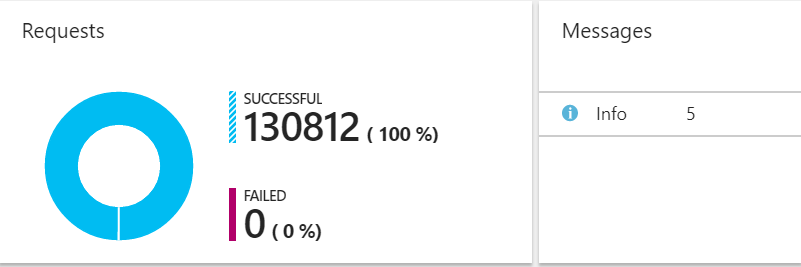


Figure 33 : Details of Test Result 1

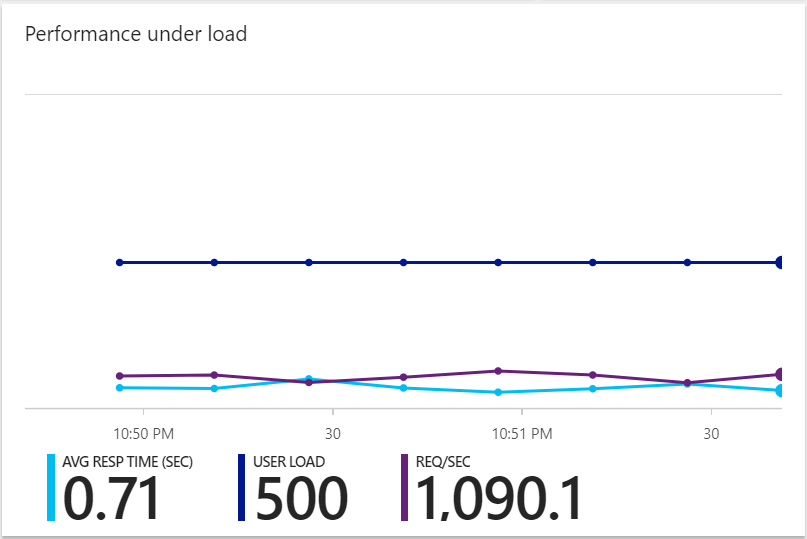


Figure 34 : Details of Test Result 2

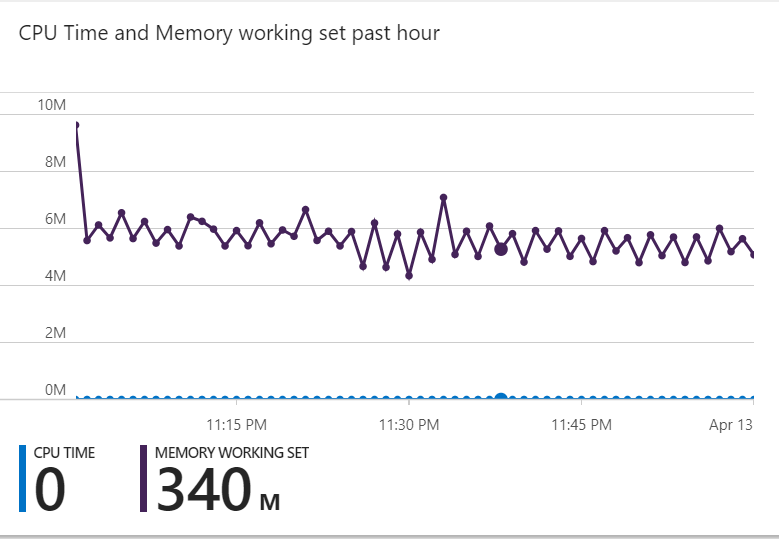


Figure 35 : Details of Test Result 3

The requests shows a 100% success rate and have an average response time of 0.71 seconds with 500 user load. The Web App Usage is shown in Figure 35.

Performance testing is further done on 750 users and 1000 users.

|  |  |  |
| --- | --- | --- |
|  | 750 Concurrent User | 1000 Concurrent User |
| Successful Request | 166685(100%) | 195400(100%) |
| Failed Request | 0 | 0 |
| Average Response Time | 1.34 | 1.53 |
| Request/second | 555.62 | 651.33 |

# 6.0 Implementation and Discussion on Managed Database (PaaS)



Figure 36 : Platform as a Service

Platform as service (PaaS) is a cloud computing model in which a third-party provider delivers hardware and software tools to users over the internet. (Apprenda, 2018) PaaS is designed to support the complete web application lifecycle: building, testing, deploying, managing and updating.

PaaS mainly provide users a resilient and optimized environment to install applications and data sets. Users can solely focus on developing and using the applications rather than constructing and maintaining the underlying infrastructure and services. (Yard, 2018) This allow users to focus to deliver the best user experience possible. The core services that PaaS provide are Mobile Software Development Kit (SDK), social and mobile built-in, rich developer environment, fully managed cloud database, point-and-click app building, multi-language development and cloud app marketplace. (Salesforce.com, 2018) Figure XX shows how an idea of a web application can be easily developed with the use of Platform as a Service.

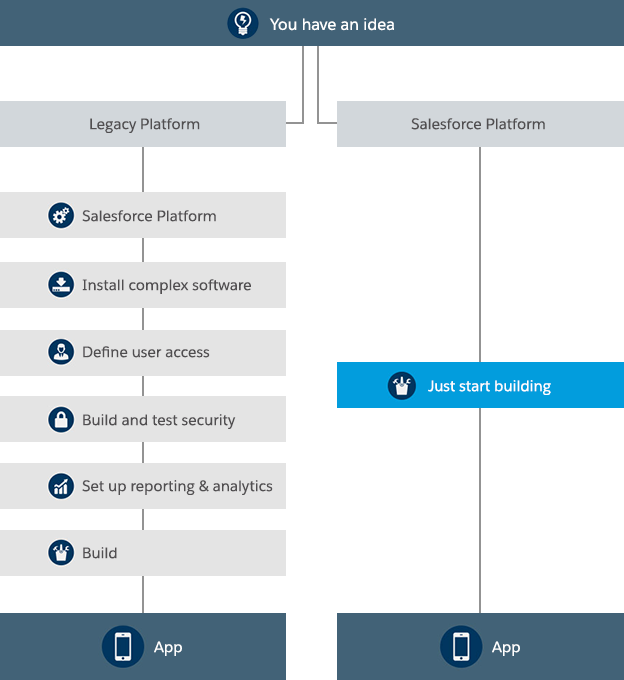


Figure 37 : Flow of creating an application

The advantages of PaaS is that it eliminates the expenses and complexity of evaluating, buying, configuring, and managing all the hardware and software needed for custom-built web applications. PaaS also cuts coding time because there are pre-coded components such as workflow, directory services and security features built in the platform. It allows expanding their business to mobile more easier because the platforms provide cross-platform services. PaaS is a pay-as-you-go model to allow the company to adjust their services as they desire depending on the size of their system. (Azure.microsoft.com, 2018)

There are various PaaS providers nowadays that supply the tools and services neede to build enterprise application in the cloud. These includes Google App Engine, Red Hat Openshift, Amazon Web Service and many more. One of the famous PaaS providers is Microsoft Azure which support application development in .NET, Node.js, PHP, Python, Java and Ruby, and allow developers to develop their application using Visual Studio and deploy them. Each PaaS has their own pricing offered for different services. (SearchCloudComputing, 2018)

# 7.0 Conclusion

The developer has gained a lot of knowledge regarding cloud computing and its architectures. The developer believes that this will be useful for the future and is used in the current business. The basic concepts of cloud computing and practiced is very vital for the completion of this assignment. From the choosing of cloud architecture, to drawing use case diagrams, class diagrams and sequence diagrams, to implementation, the developer has gained a lot of experience and valuable help from friends whenever problems occurred. Cloud is about how you do computing, not where you do computing. The developer believes the deployment process and use of cloud computing is useful for future career and is grateful to Dr. Kalai.

# References

Apprenda. (2018). *IaaS, PaaS, SaaS (Explained and Compared)*. [online] Available at: https://apprenda.com/library/paas/iaas-paas-saas-explained-compared/ [Accessed 12 Apr. 2018].

Azure.microsoft.com. (2018). *What is PaaS? Platform as a Service | Microsoft Azure*. [online] Available at: https://azure.microsoft.com/en-us/overview/what-is-paas/ [Accessed 12 Apr. 2018].

Docs.microsoft.com. (2018). *What is Traffic Manager*. [online] Available at: https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview [Accessed 12 Apr. 2018].

Salesforce.com. (2018). *PaaS*. [online] Available at: https://www.salesforce.com/ap/learning-centre/tech/paas/ [Accessed 12 Apr. 2018].

SearchCloudComputing. (2018). *What is Platform as a Service (PaaS)? - Definition from WhatIs.com*. [online] Available at: https://searchcloudcomputing.techtarget.com/definition/Platform-as-a-Service-PaaS [Accessed 12 Apr. 2018].

Yard, E. (2018). *What is Platform as a Service (PaaS) Cloud Computing? | Engine Yard*. [online] Engineyard.com. Available at: https://www.engineyard.com/platform-as-a-service-cloud [Accessed 12 Apr. 2018].