Table of Contents

[1.0 Introduction 3](#_Toc511406970)

[1.1 Background 3](#_Toc511406971)

[1.2 Objective 3](#_Toc511406972)

[1.3 Scope 3](#_Toc511406973)

[1.4 Functional Requirement Specification 3](#_Toc511406974)

[1.5 Non-functional Requirement Specification 4](#_Toc511406975)

[2.0 Project Plan 4](#_Toc511406976)

[3.0 Design Implementation 6](#_Toc511406977)

[3.1 Cloud Architecture 6](#_Toc511406978)

[3.2 Data Consideration 7](#_Toc511406979)

[3.3 UML Modelling 7](#_Toc511406980)

[3.3.1 Use Case Diagram 7](#_Toc511406981)

[3.3.3 Sequence Diagram 14](#_Toc511406982)

[4.0 Implementation 18](#_Toc511406983)

[4.1 ASP.NET Web Application 18](#_Toc511406984)

[4.2 Resource Group in Azure 18](#_Toc511406985)

[4.2.1 Resource Group Web App Service 18](#_Toc511406986)

[4.2.3 SQL Database Azure 19](#_Toc511406987)

[4.2.4 Resource Group Traffic Manager 22](#_Toc511406988)

[4.3 Geo-Replication 23](#_Toc511406989)

[4.3.1 Server created Geo-Replication 23](#_Toc511406990)

[4.4 Creating App Service using Web App+SQL 24](#_Toc511406991)

[4.4.1 Creating app service plan-Southeast Asia 24](#_Toc511406992)

[4.4.2 Creating app service plan -UK West 25](#_Toc511406993)

[4.5 Traffic Manager 26](#_Toc511406994)

[4.6 Traffic Manager setting endpoint 27](#_Toc511406995)

[4.6.1 Endpoint set 27](#_Toc511406996)

[4.7 Deployment – Publish System to Online 28](#_Toc511406997)

[4.7.1 Setting up Admin username and password 28](#_Toc511406998)

[4.8 Published Successful – Southeast Asia 29](#_Toc511406999)

[4.8.1 Published Successful – UK West 29](#_Toc511407000)

[5.0 System Interfaces 32](#_Toc511407001)

[5.1 Main Page 32](#_Toc511407002)

[5.2 Login Page – Login as Admin 32](#_Toc511407003)

[5.3 Home Page 32](#_Toc511407004)

[5.4 Schedule Page 33](#_Toc511407005)

[5.5 Ship Page 33](#_Toc511407006)

[5.6 Customer Page 34](#_Toc511407007)

[5.7 Booking Page 34](#_Toc511407008)

[5.8 Register Agent Page 34](#_Toc511407009)

[5.9 Unit Testing 35](#_Toc511407010)

[6.0 Conclusion 35](#_Toc511407011)

[Reference 36](#_Toc511407012)

[Appendix 37](#_Toc511407013)

# 1.0 Introduction

## 1.1 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 which 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. At the point, as the company gain big, Maersk Line has reached the point to decide to use cloud as the solution.

The purpose Maersk Line using cloud application is because cloud application can use to support current and future growth of the company business and increase flexibility as well as efficiency. The organization decided to consolidate their data centres and improved server around the world to the virtual platform. Thus, Mircosoft Azure was chosen to be the host of the IT environment and platform to expand their relationship. Firstly, the changing over IT setup by using Azure started with the container management.

## 1.2 Objective

The project aims to provide a solution that to reduce the organization overall chain costs and improve logistic operation efficiency by developing a efficient cloud-based application to accelerate the business process.

## 1.3 Scope

The project included the process of developing and designing a cloud-based system for the Container Management System (CMS) by hosting the system to the Microsoft Azure platform, allowing users to access from different geographical areas.

## 1.4 Functional Requirement Specification

Maersk Line Container Management System (CMS) use cloud-based application that is compatible with the most used web browsers including Google Chrome, Microsft Ege and Mozilla Firefox. The functional requirement specification of the system accessible by the user is as table below.

|  |  |
| --- | --- |
| User | Functional Requirement |
| Admin | * Register Agent * Login * Create, edit, view schedule * Create, edit, delete, view ship * Create, edit, delete, view customer * Create edit, view booking * Logout |
| Agents | * Login * Create, edit, delete, view customer * Create and view Booking * Logout |

*Table 1: Functional Requirement of CMS*

## 1.5 Non-functional Requirement Specification

1. Provisioning

* Able to provision the new application to the Microsoft Azure Platform.

2. Maintainability

* Able to upgrade the application and perform maintenance while multiple users are using it.

3. Monitoring

* Able to monitor the application at any time to identify errors occurs and troubleshoot them.

4. Availability

* The web application constantly available to be accessed.

5. Scalability

* The application should be scalable to meets the requirement of the application.

# 2.0 Project Plan

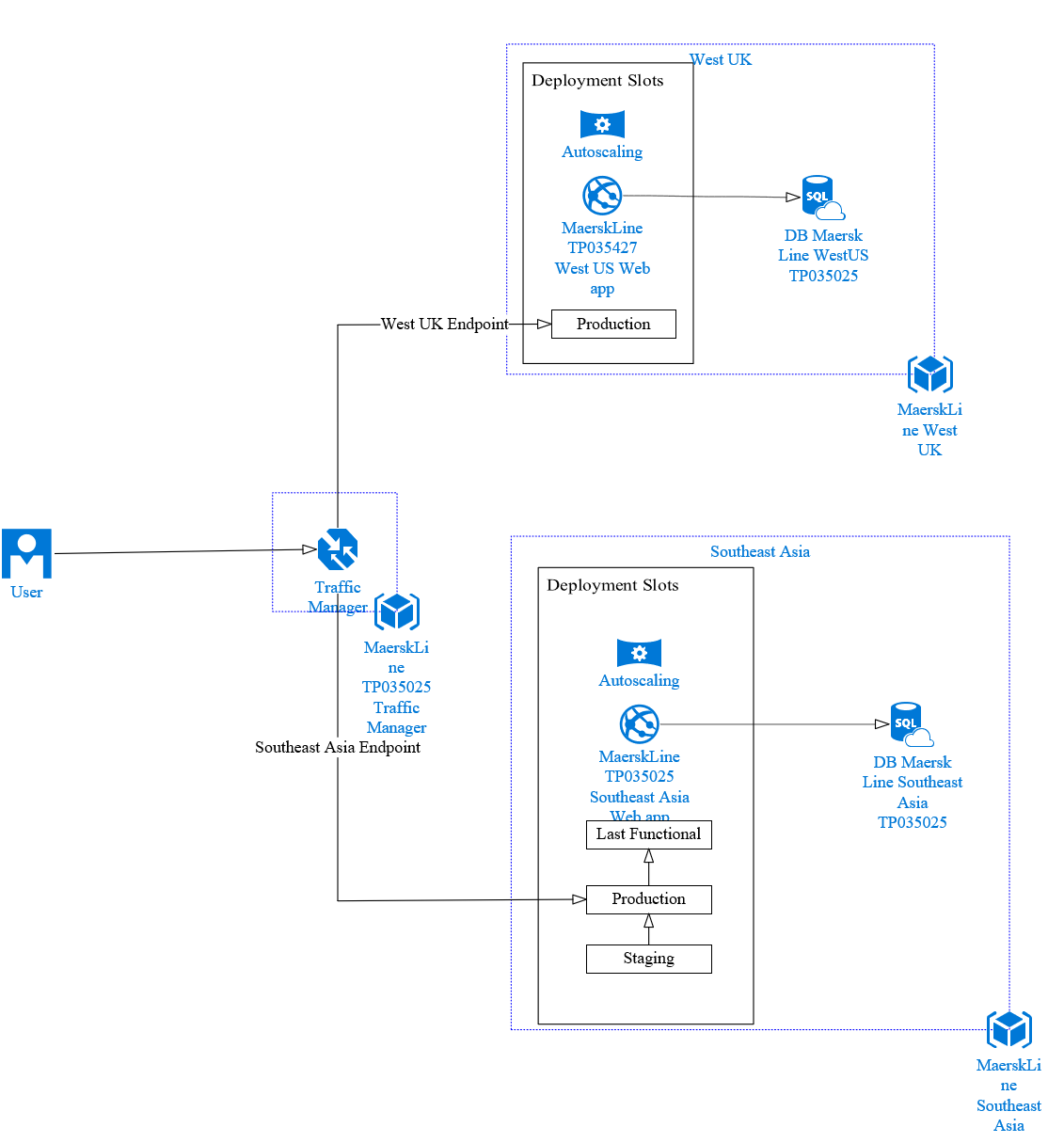
2.1 Work Breakdown Structure (WBS)

|  |  |  |
| --- | --- | --- |
| WBS | Task Name | Duration |
| **1** | **Maersk Line CMS** | **28 days** |
| **1.1** | **Introduction** | **2 days** |
| 1.1.1 | Background | 1 day |
| 1.1.2 | Objective | 1 day |
| 1.1.3 | Scope | 1 day |
| 1.1.4 | Non-functional Requirement Specification | 1 day |
| 1.1.5 | Functional Requirement Specification | 1 day |
| **1.2** | **Project Plan** | **1 day** |
| 1.2.1 | Work Breakdown Structure | 1 day |
| 1.2.2 | Gantt Chart | 1 day |
| **1.3** | **Design** | **7 days** |
| 1.3.1 | Cloud Architecture | 1 day |
| 1.3.2 | Design Consideration | 1 day |
| **1.3.3** | **Modelling** | **6 days** |
| 1.3.3.1 | Use Case Diagram | 1 day |
| 1.3.3.2 | Use Case Description | 3 days |
| 1.3.3.3 | Sequence Diagram | 3 days |
| **1.4** | **Implementation** | **13 days** |
| 1.4.1 | ASP.NET Web Application | 10 days |
| 1.4.2 | Azure Resource Group | 1 day |
| 1.4.3 | Azure SQL Server | 1 day |
| 1.4.4 | Azure SQL Database | 1 day |
| 1.4.5 | Azure Web Application Service | 1 day |
| 1.4.6 | Azure Traffic Manager | 1 day |
| 1.4.7 | Azure Web Application Autoscaling | 1 day |
| 1.4.8 | System Interface | 3 days |
| **1.5** | **Testing** | **2 days** |
| 1.5.1 | Performance Testing | 1 day |
| 1.5.2 | Unit Testing | 1 day |
| 1.6 | Conclusion | 1 day |

*Table 2: WBS*

# 3.0 Design Implementation

## 3.1 Cloud Architecture



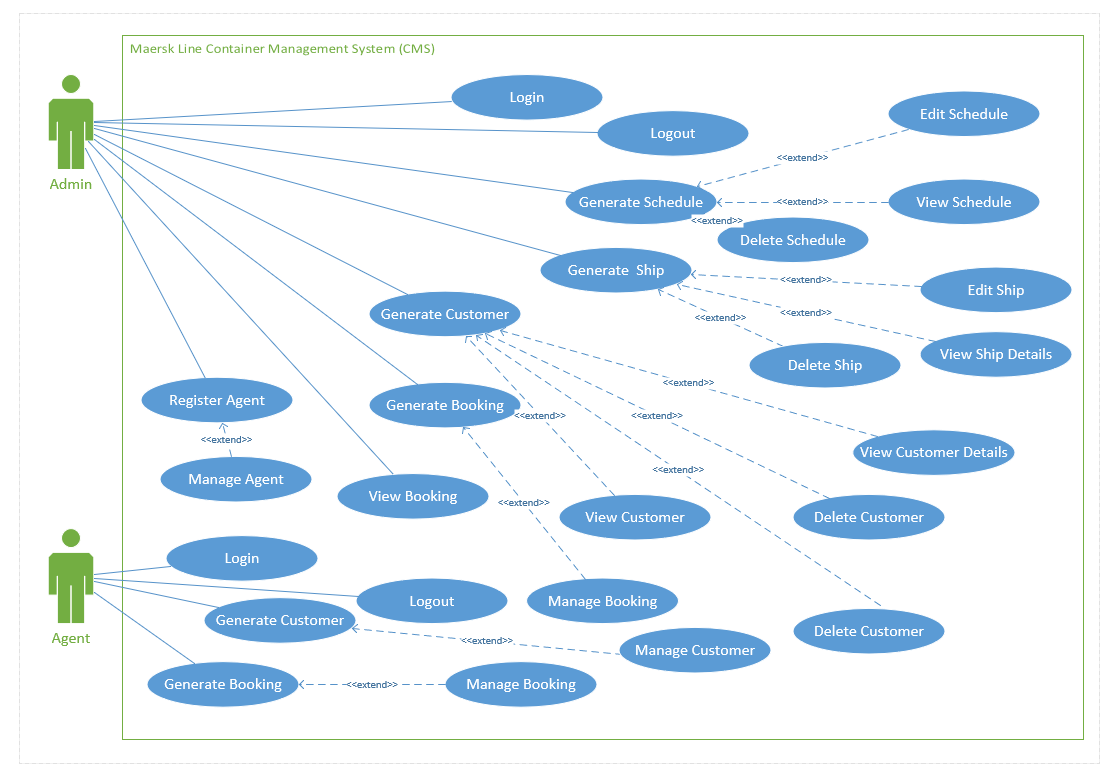
The figure above showed the cloud architecture that used for deploying the CMS of Maersk Line company to the Microsoft Azure cloud platform. The deployment adopts two-region designs, which primary region cloud services will be in Southeast Asia and secondary located at West Europe. Meanwhile, each region has an individual web app and using autoscaling service up that support 1 to 3 instances. The individual web apps are monitor by the traffic manager at both regions. Besides that, each of the region have its own database to ensure the quality and performance of the application and database is nearer to each other.

## 3.2 Data Consideration

There are several considerations that needed to be plan and discuss over before proceeding to the design phase. The main purpose of Maersk Line wanted to use the cloud application as solution is that to reduce supply chain budget and can have a efficient way to manage their company logistic. Hence, a high-performance cloud application is developed that able to cover one or more region that able to access the system. At the same time, the development team facing limited budget of Microsoft Azure cloud platform credit therefore the team decided to host the system to Southeast Asia and West Europe region only to save cost for the hosting fee.

## 3.3 UML Modelling

### 3.3.1 Use Case Diagram



Use Case Specification

Table 1 - Login

|  |  |
| --- | --- |
| Use Case ID: | 1 |
| Use Case Name: | Login |
| Summary: | Users able to login to the system. |
| Dependency: | N/A |
| Actors: | Admin/Agent |
| Precondition: | User requires valid username and password in order to login to the web application. |
| Description Flow: | 1. Login page is displayed. 2. The system prompts user to fill in username and password. 3. The user entered username and password 4. System verify user login credential. 5. User logged in to the web application successfully. |
| Alternative Sequence: | * 1. If login credentials are valid, then the user login in to the system successfully.   2. If login credentials are invalid, then the system prompts users to login page. |
| Post-condition | The user able access to the system. |

Table 2: Create Shipment

|  |  |
| --- | --- |
| Use Case ID: | 2 |
| Use Case Name: | Logout |
| Summary: | User logout from the system |
| Dependency: | N/A |
| Actors: | Admin/Agent |
| Precondition: | User login to the system. |
| Description Flow: | 1. User login to the system successfully. 2. User able to see home page. 3. User able to logout the system. |
| Alternative Sequence: | 1. If the user failed to login the system, then the user cannot logout. |
| Post-condition | User logout the system. |

Table 3: Register Agent

|  |  |
| --- | --- |
| Use Case ID: | 3 |
| Use Case Name: | Register Agent |
| Summary: | Admin register new agent to the system |
| Dependency: | Manage Agent |
| Actors: | Admin |
| Precondition: | User login to the system as Admin. |
| Description Flow: | 1. User login to the system as Admin.  2. User able to select register agent from the navigation bar.  3. User able to register new agent. |
| Alternative Sequence: | 1. If the information entered is invalid, error message will display. |
| Post-condition | Agent account successfully registered. |

Table 4: Generate Schedule

|  |  |
| --- | --- |
| Use Case ID: | 4 |
| Use Case Name: | Generate Schedule |
| Summary: | Admin generate new Schedule record. |
| Dependency: | <<extend>> Create Schedule  <<extend>> Edit Schedule  <<extend>> Delete Schedule  <<extend>> View Schedule |
| Actors: | Admin |
| Precondition: | User login to the system as Admin. |
| Description Flow: | Generate Schedule  1. User login to the system as Admin.  2. User able to select schedule from the navigation bar.  3. User prompted to the schedule page.  4. User enter schedule information.  5. New Schedule created successfully.  Edit Schedule  1. Select a schedule record to edit.  2. Schedule Edit successfully.  View Schedule  1. Select a schedule record to view.  2. System display the specific schedule information  Delete Schedule  1. Select a schedule record to delete.  2. Schedule record deleted successfully. |
| Alternative Sequence: | Invalid Schedule information entered. |
| Post-condition | - |

Table 5: Generate Ship

|  |  |
| --- | --- |
| Use Case ID: | 5 |
| Use Case Name: | Generate Ship |
| Summary: | Admin generate new ship record. |
| Dependency: | <<extend>> Edit Ship  <<extend>> Delete Ship  <<extend>> View Ship |
| Actors: | Admin |
| Precondition: | User login to the system as Admin. |
| Description Flow: | Generate Ship  1. User login to the system as Admin.  2. User able to select ship from the navigation bar.  3. User prompted to the ship page.  4. User enter ship information.  5. New Ship record created successfully.  Edit Ship  1. Select a ship record to edit.  2. Ship record edited successfully.  View Ship  1. Select a ship record to view.  2. System display the specific ship information  Delete Ship  1. Select a ship record to delete.  2. Ship record deleted successfully. |
| Alternative Sequence: | Invalid Ship information entered |
| Post-condition | - |

Table 6: Generate Customer

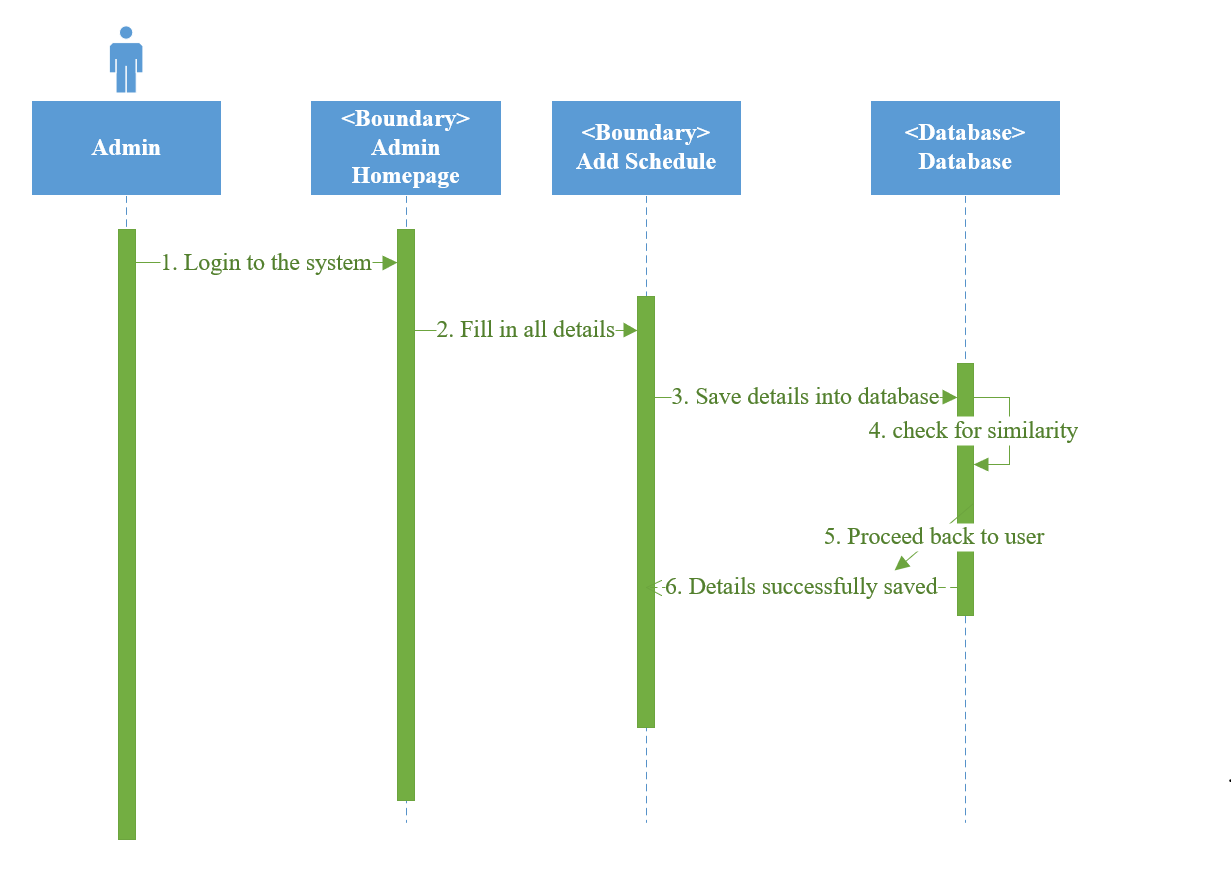
|  |  |
| --- | --- |
| Use Case ID: | 6 |
| Use Case Name: | Generate Customer |
| Summary: | Admin register new customer record. |
| Dependency: | <<extend>> Edit Customer  <<extend>> Delete Customer  <<extend>> View Customer |
| Actors: | Admin/Agent |
| Precondition: | User login to the system as Admin. |
| Description Flow: | Generate Customer  1. User login to the system as Admin.  2. User able to select customer from the navigation bar.  3. User prompted to the customer page.  4. User enter customer information.  5. New customer record created successfully.  Edit Customer  1. Select a customer record to edit.  2. Customer record edited successfully.  View Customer Detail  1. Select a customer record to view.  2. System display customer information.  Delete Customer  1. Select a customer record to delete.  2. Customer record deleted successfully. |
| Alternative Sequence: | Invalid Customer Information entered. |
| Post-condition | - |

Table 7: Generate Booking

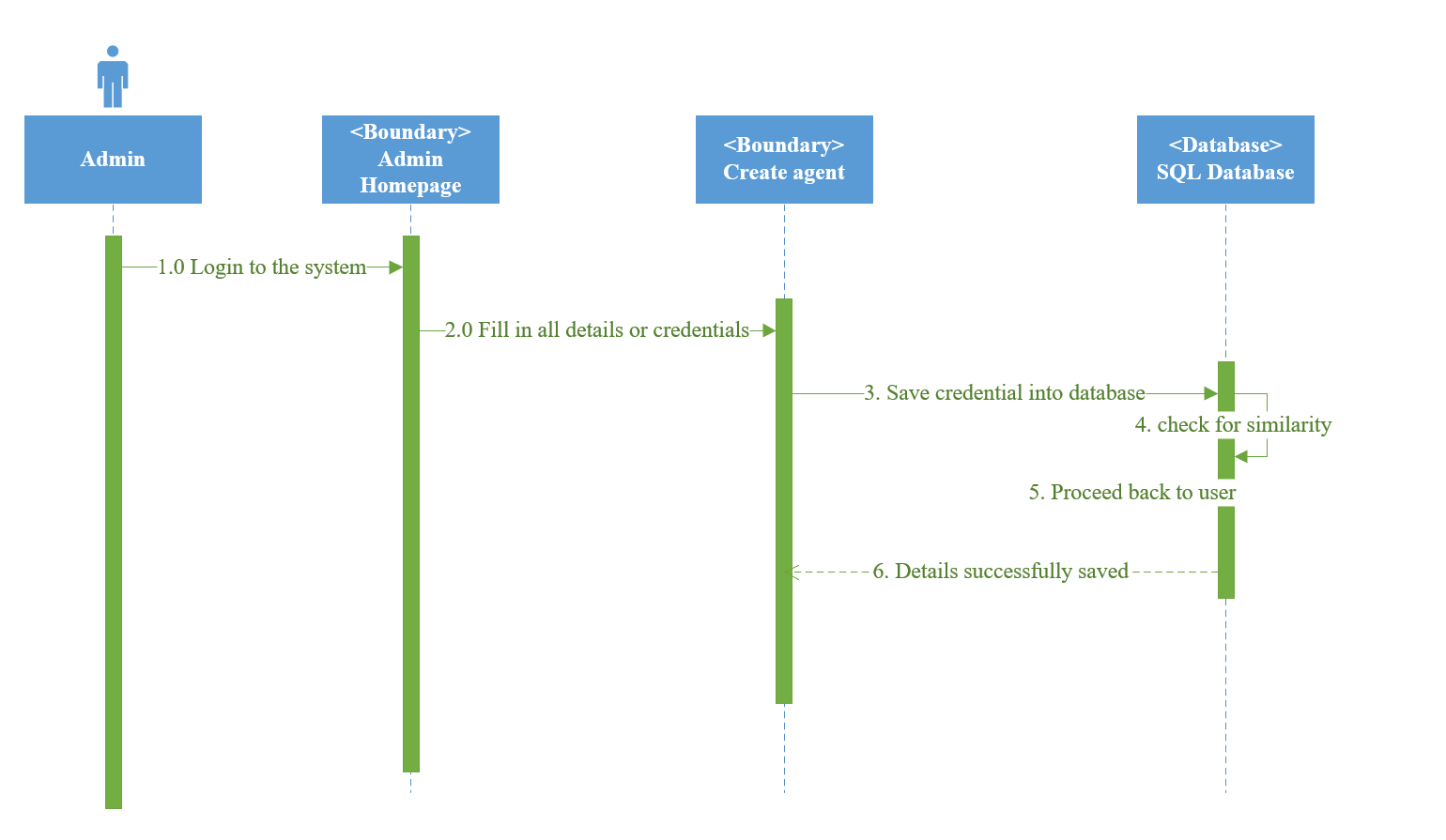
|  |  |
| --- | --- |
| Use Case ID: | 7 |
| Use Case Name: | Generate Booking |
| Summary: | Admin register new customer record. |
| Dependency: | <<extend>> Manage Booking |
| Actors: | Admin/Agent |
| Precondition: | User login to the system as Admin. |
| Description Flow: | Generate Booking  1. User login to the system as Admin.  2. User able to select booking from the navigation bar.  3. User prompted to the booking page.  4. User enter booking information.  5. New booking record created successfully.  Manage Booking  1. Admin manage the booking after the booking record created. |
| Alternative Sequence: | Invalid booking information entered. |
| Post-condition | - |

### 3.3.3 Sequence Diagram

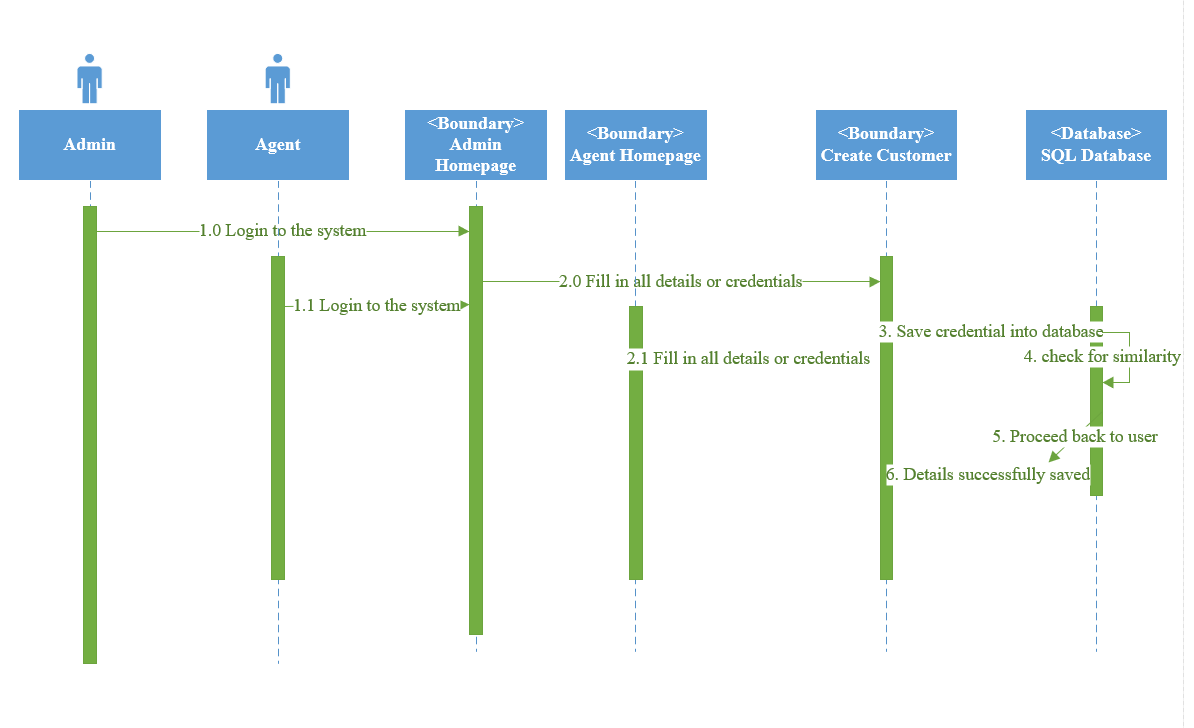
#### 3.3.3.1 Add Schedule



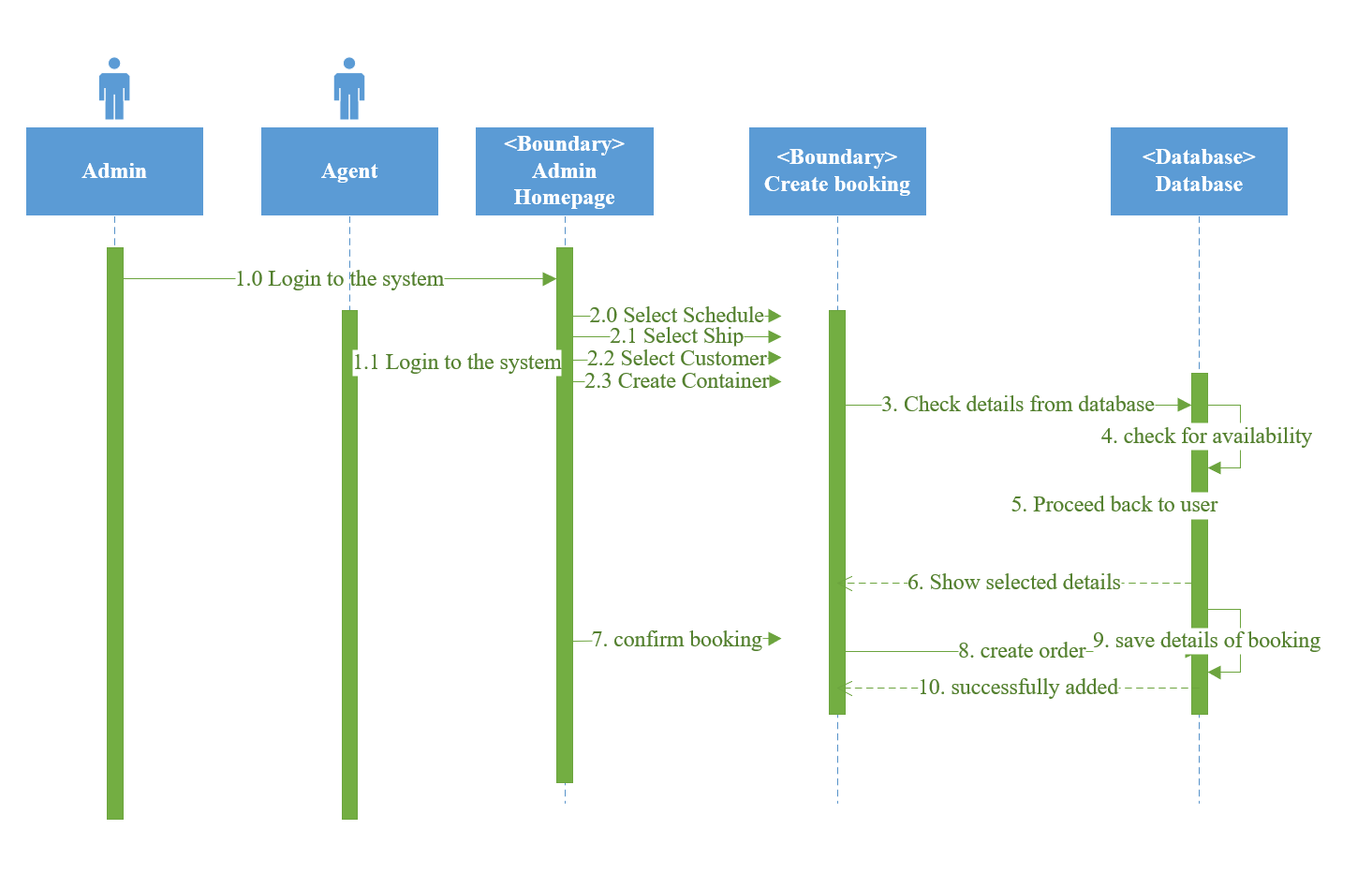
#### 3.3.3.2 Create Agent



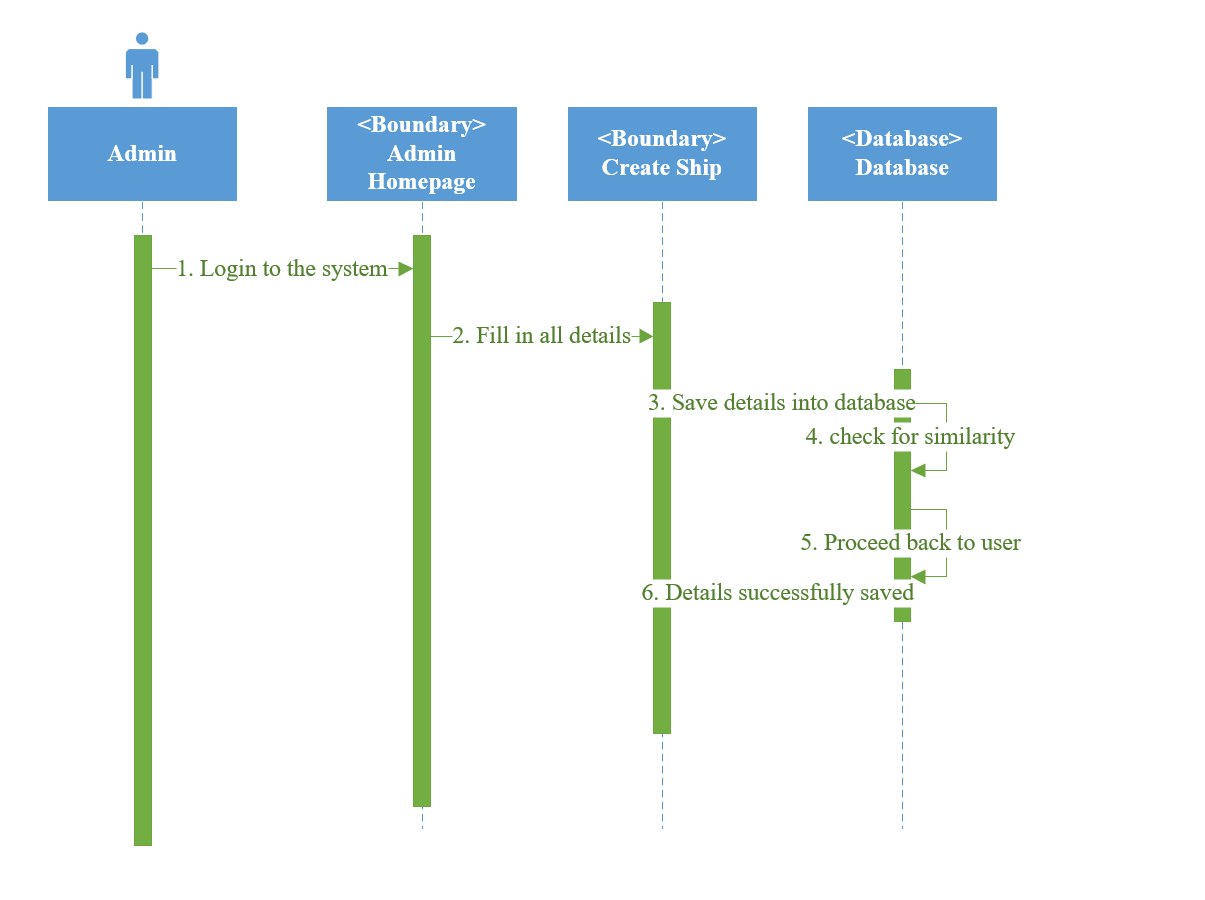
#### 3.3.3.4 Create Customer



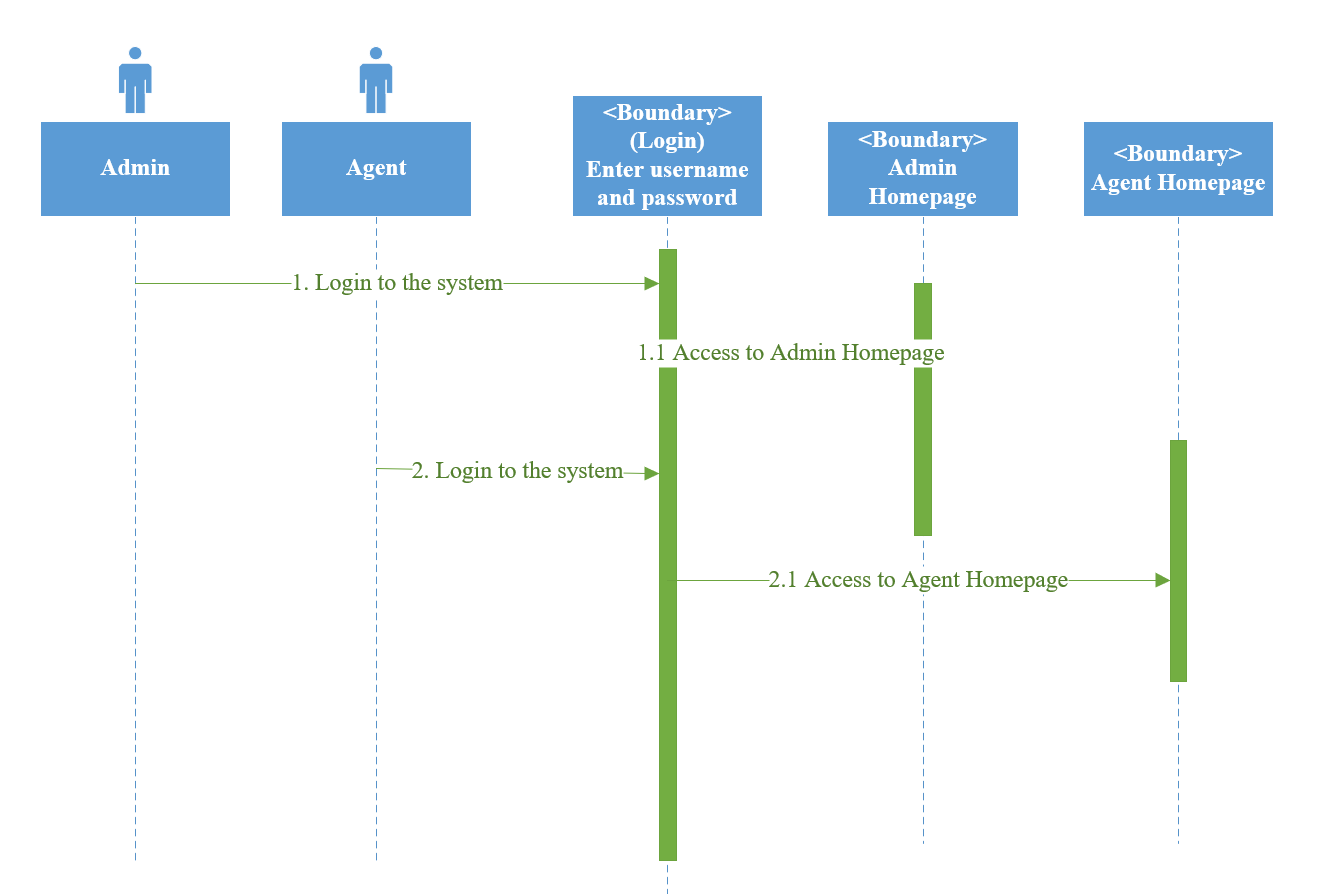
#### 3.3.3.5 Create Booking



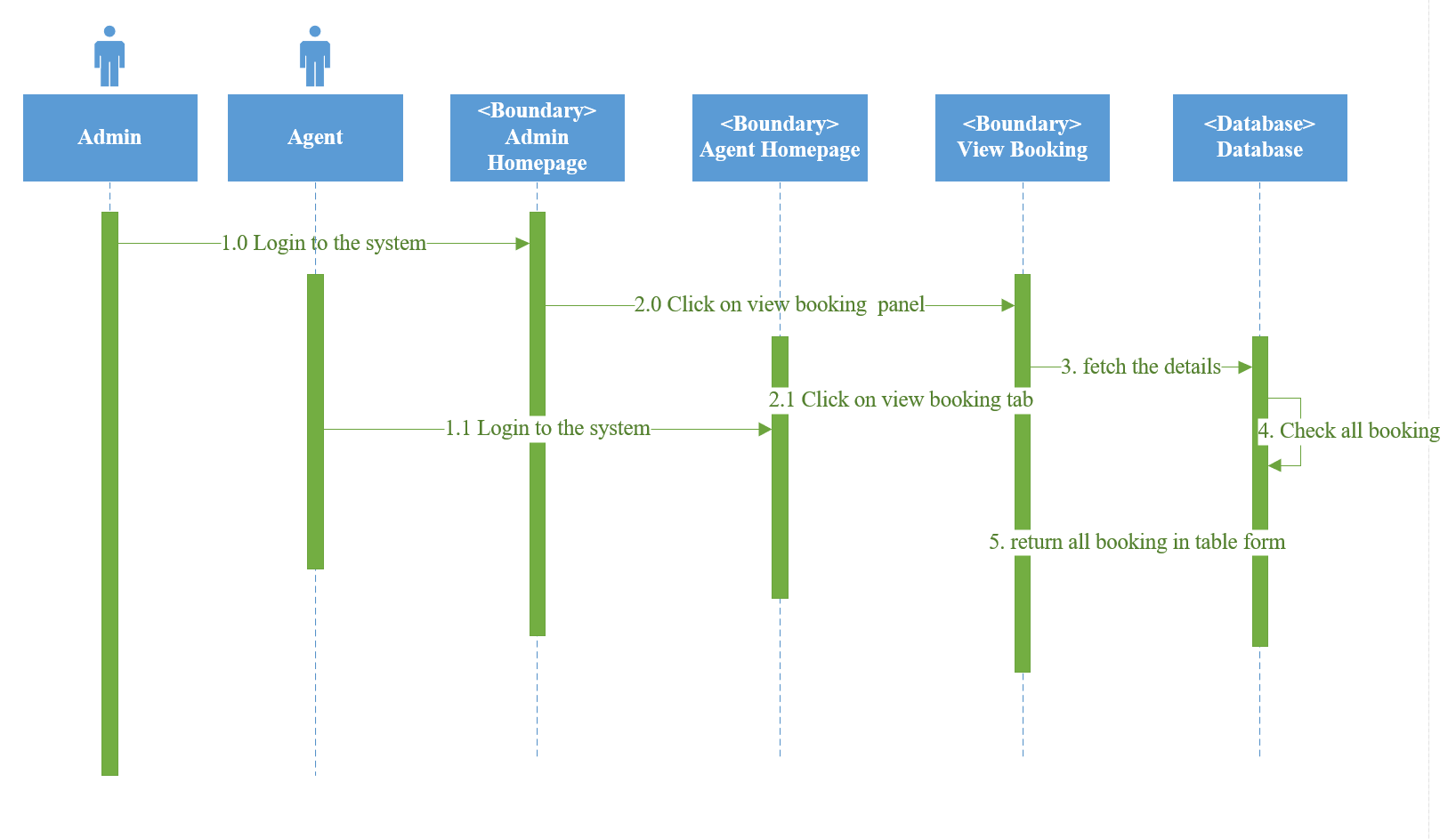
#### 3.3.3.6 Create Ship



#### 3.3.3.7 Login



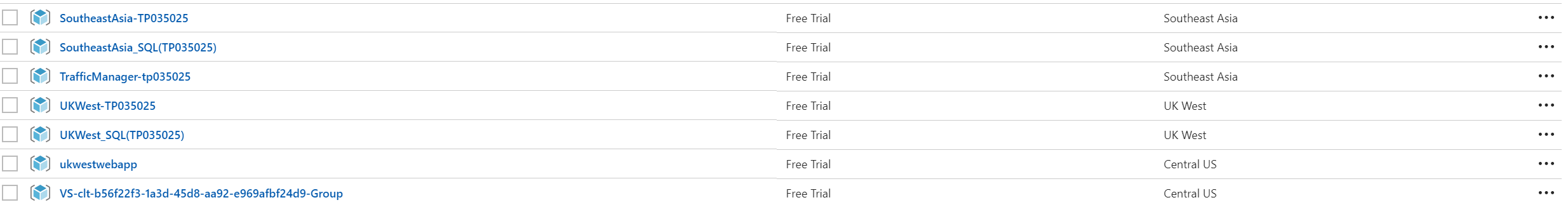
#### 3.3.3.8 View Booking



# 4.0 Implementation

## 4.1 ASP.NET Web Application

## 4.2 Resource Group in Azure

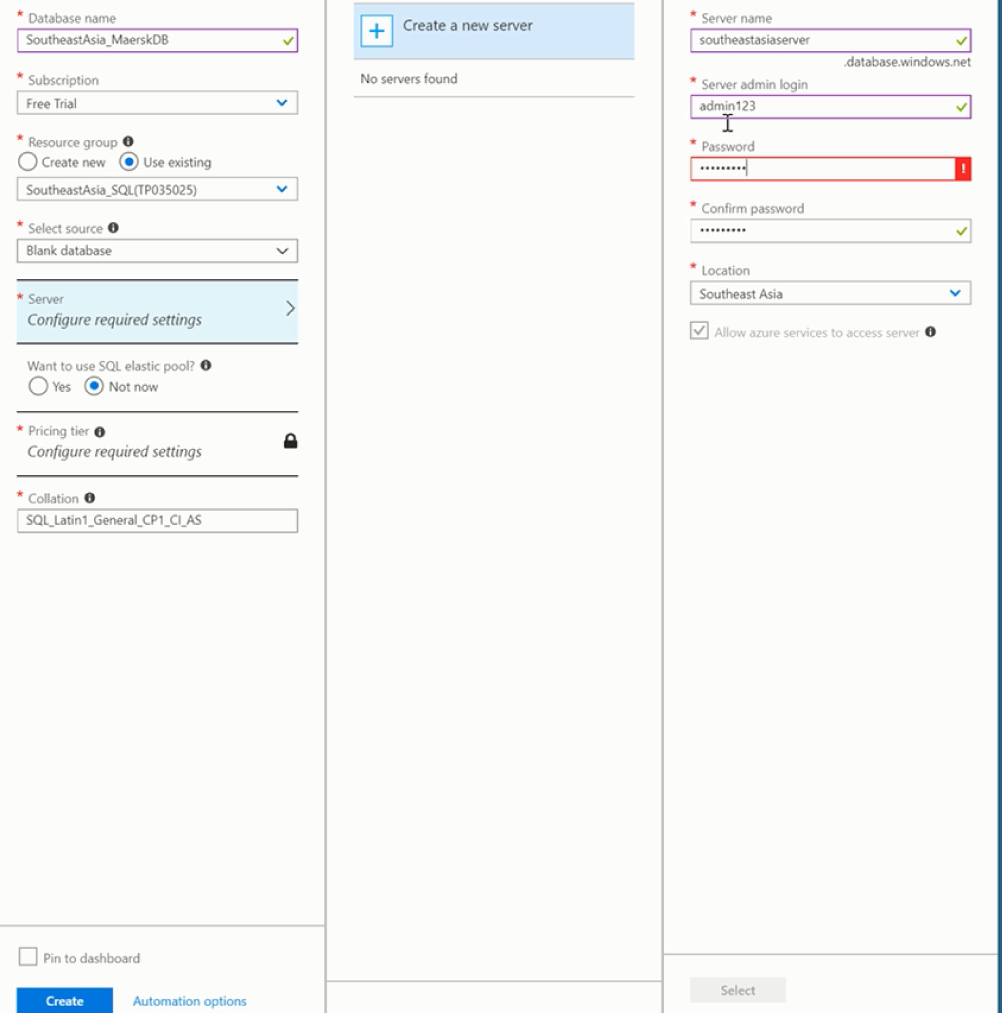


### 4.2.1 Resource Group Web App Service

### 4.2.3 SQL Database Azure

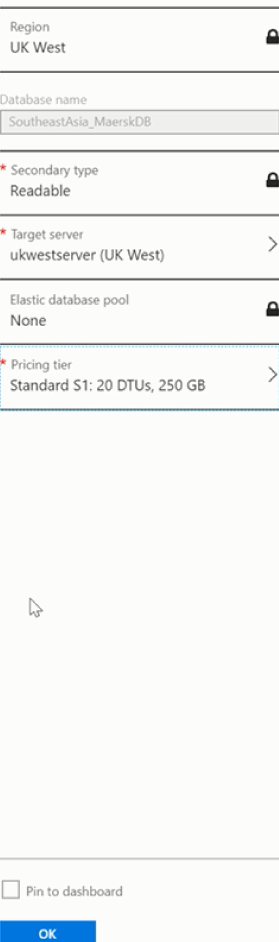
#### 4.2.3.1 Creating SoutheastAsia\_MaerskDB



#### 4.2.3.2 Using SoutheastAsia Geo-Replication



Using Geo-Replication to create UK West SQL Database

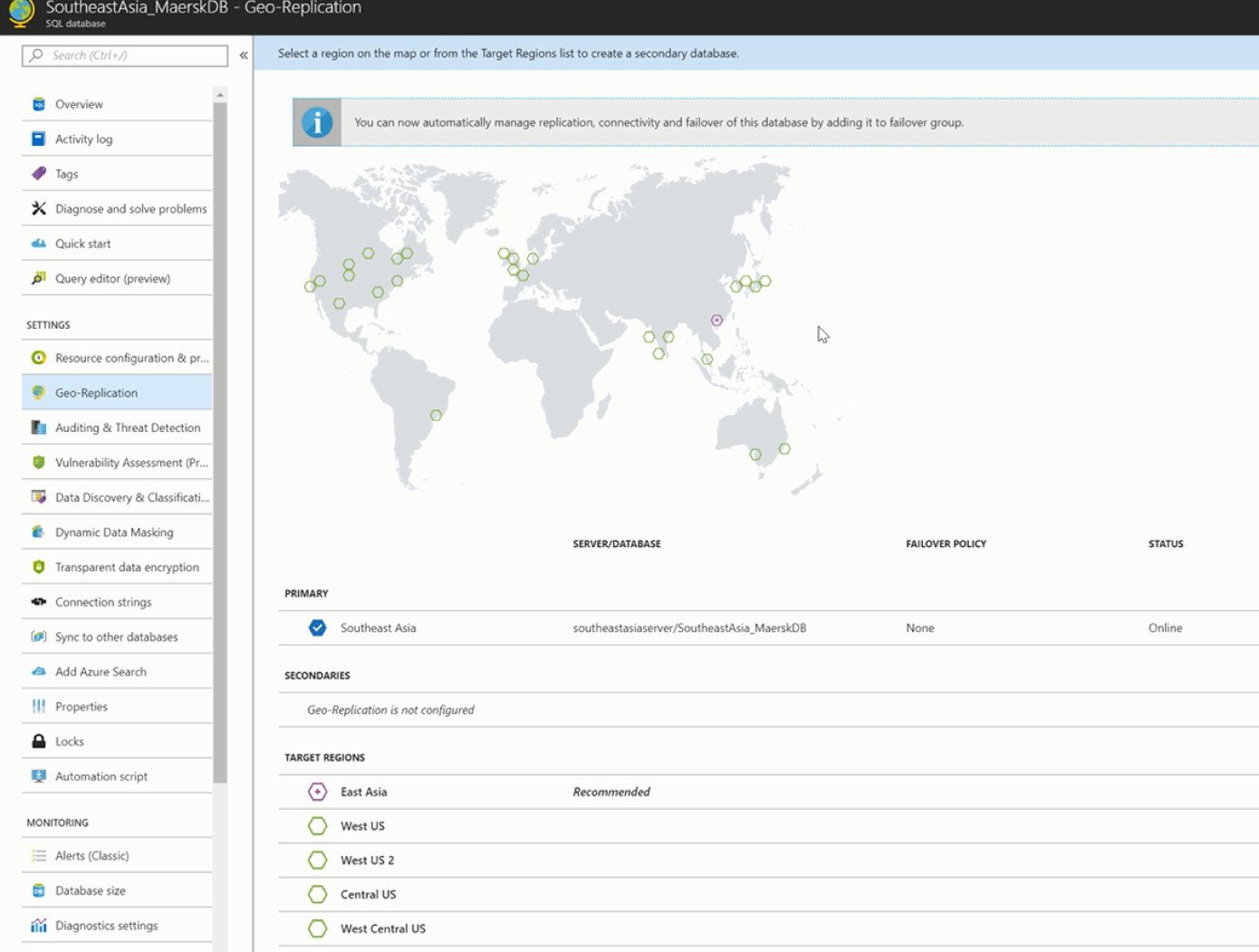


### 4.2.4 Resource Group Traffic Manager

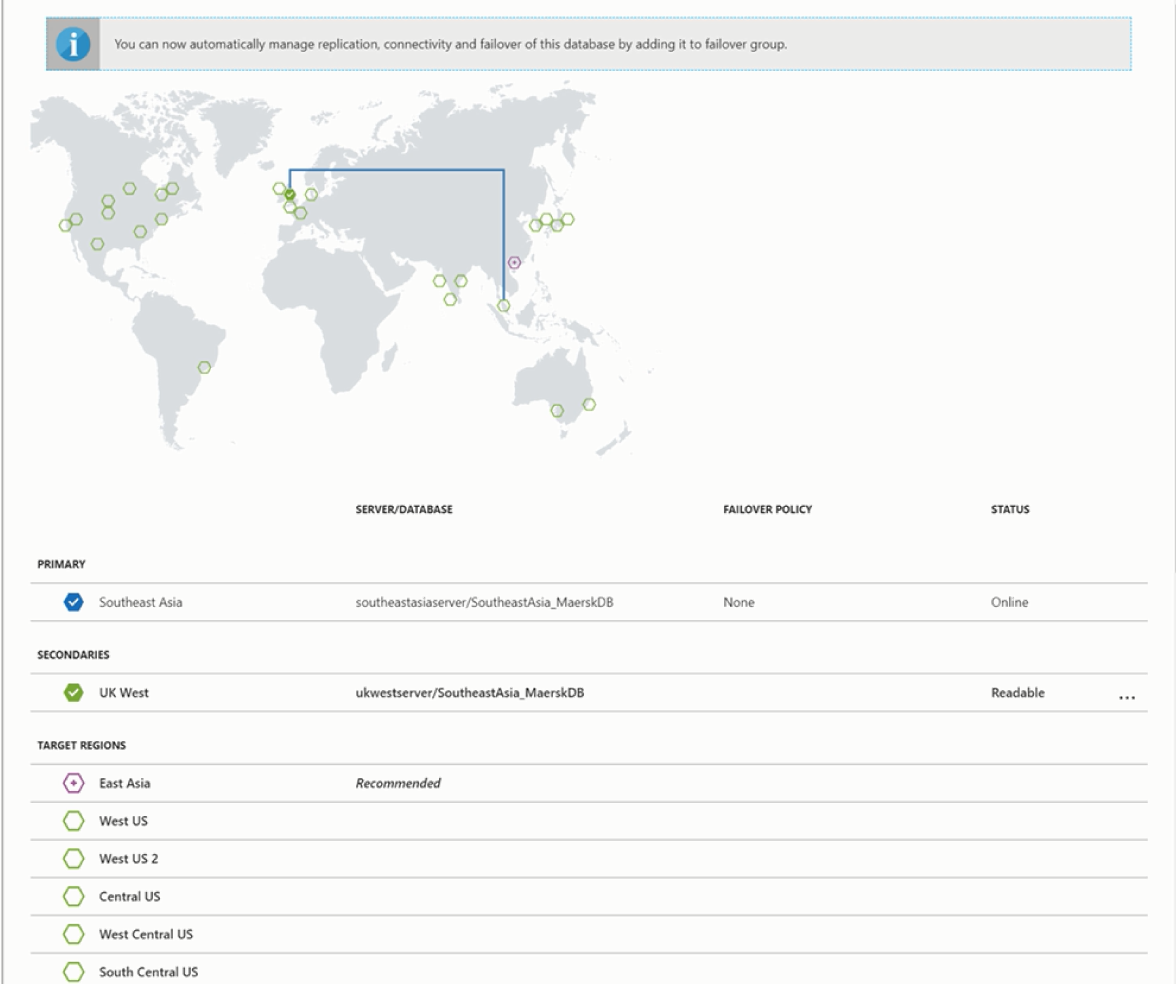
A traffic manager resource group created with the name “SoutheastAsia-TrafficManager(TP035025).



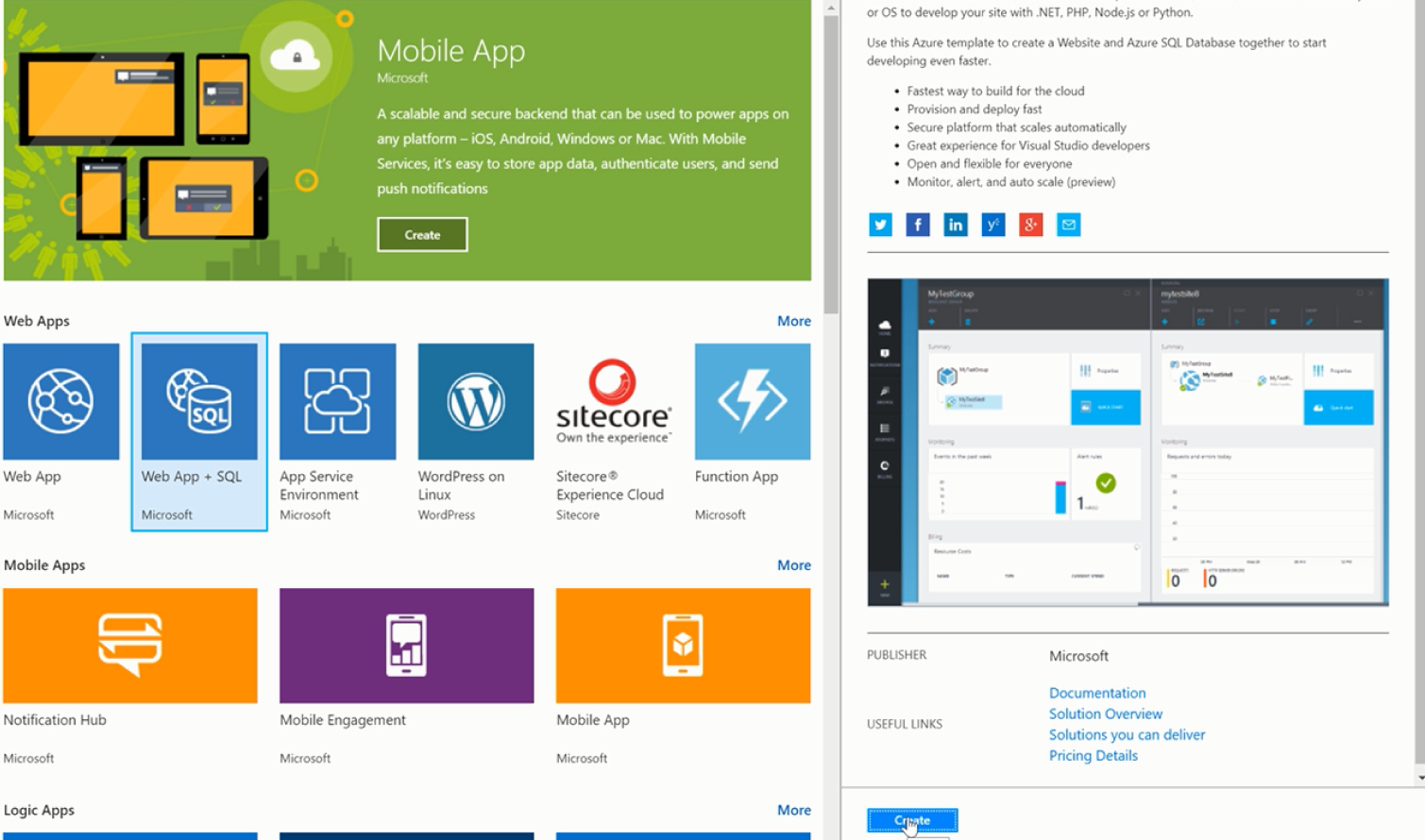
## 4.3 Geo-Replication



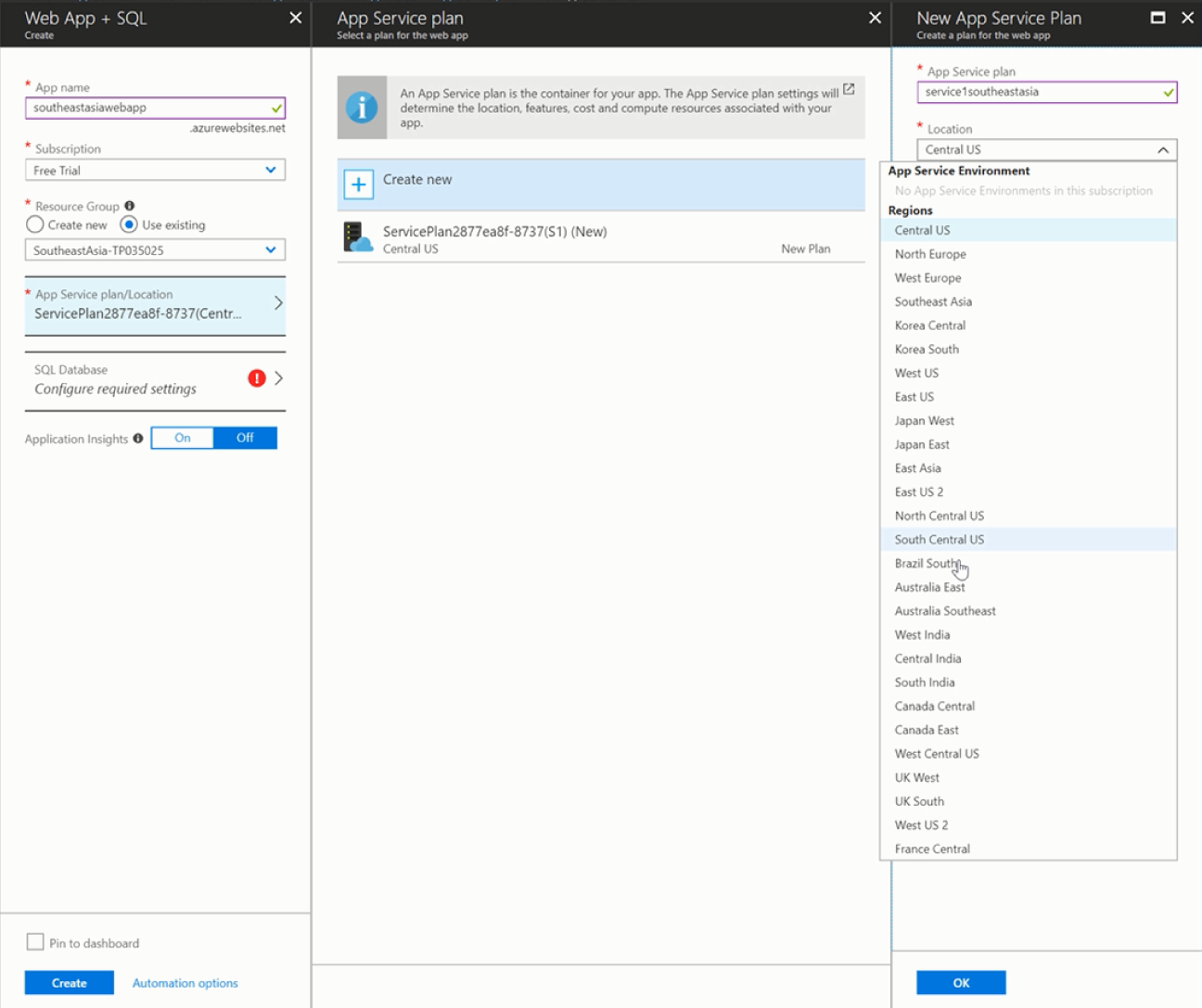
### 4.3.1 Server created Geo-Replication



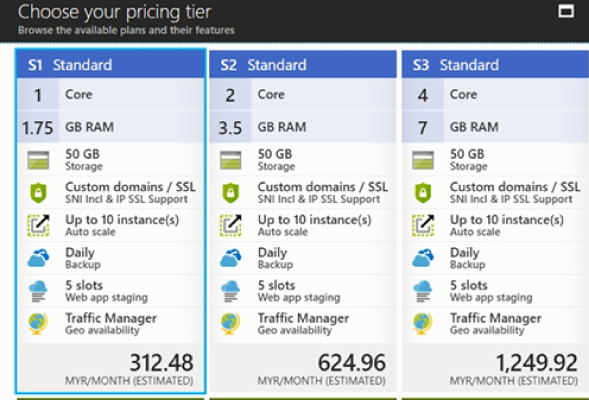
## 4.4 Creating App Service using Web App+SQL



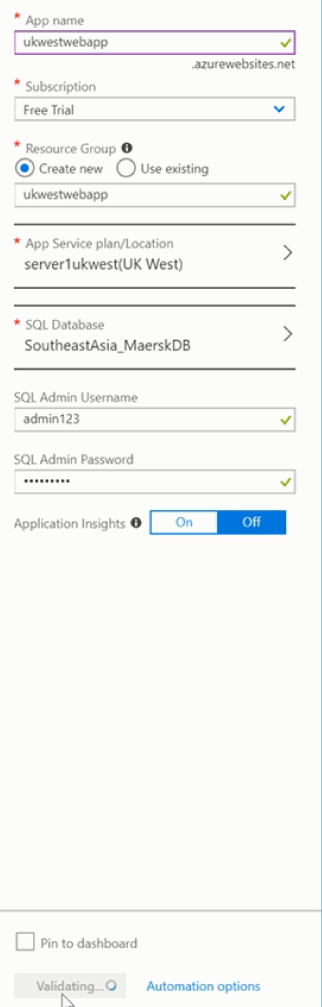
### 4.4.1 Creating app service plan-Southeast Asia



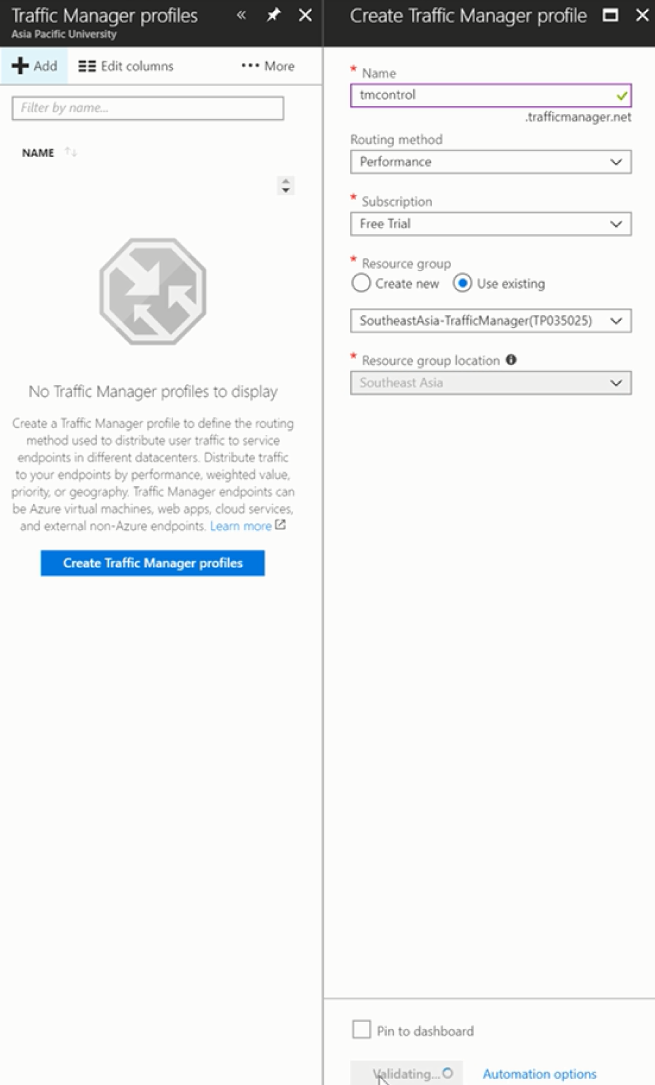
#### 4.4.1.1 Using the S1 standard pricing tier.



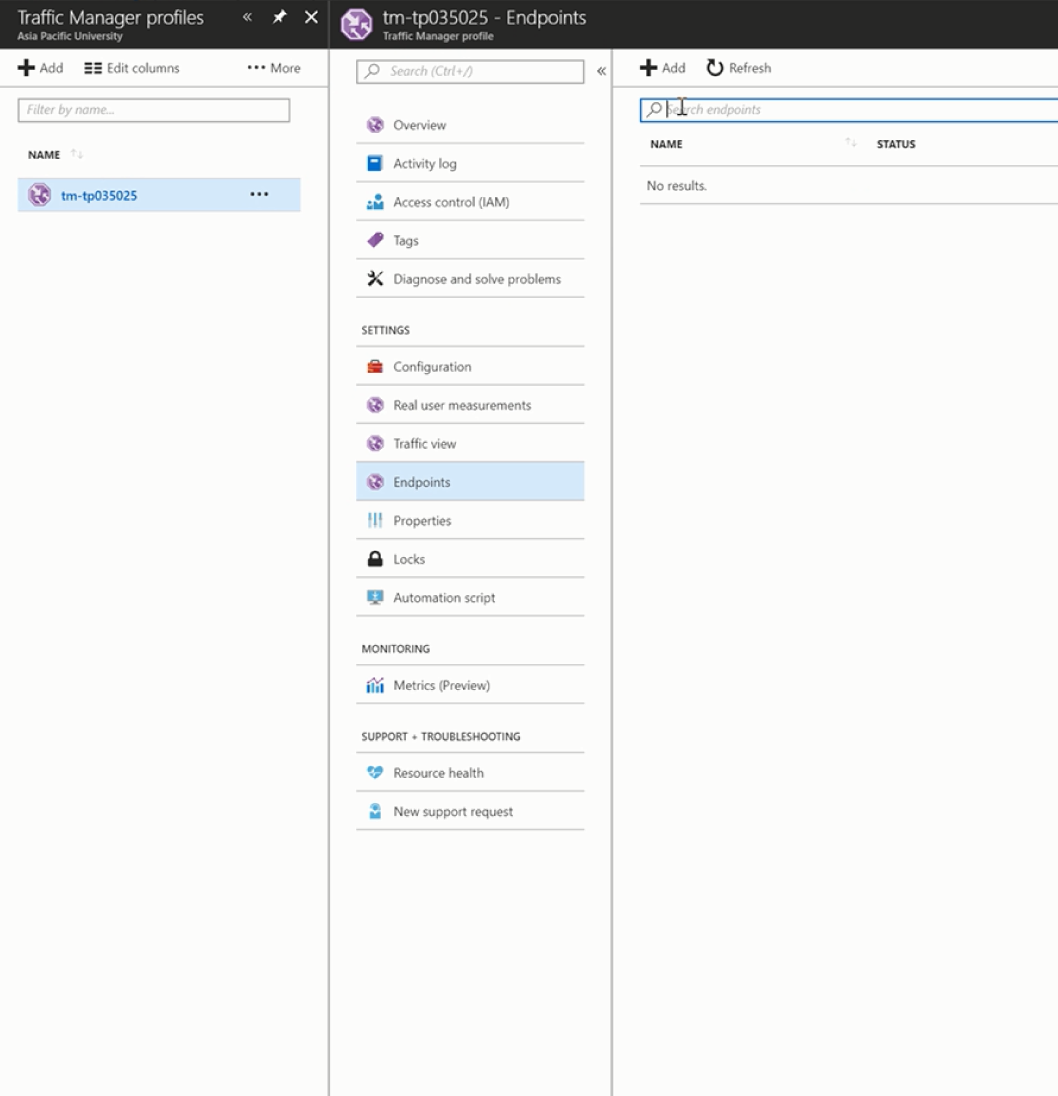
### 4.4.2 Creating app service plan -UK West



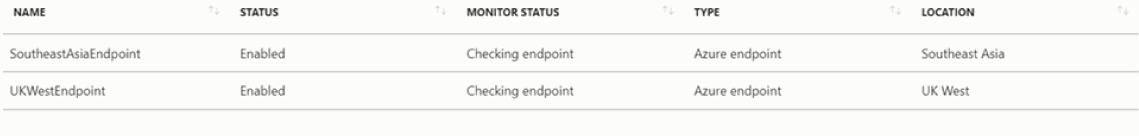
## 4.5 Traffic Manager



## 4.6 Traffic Manager setting endpoint

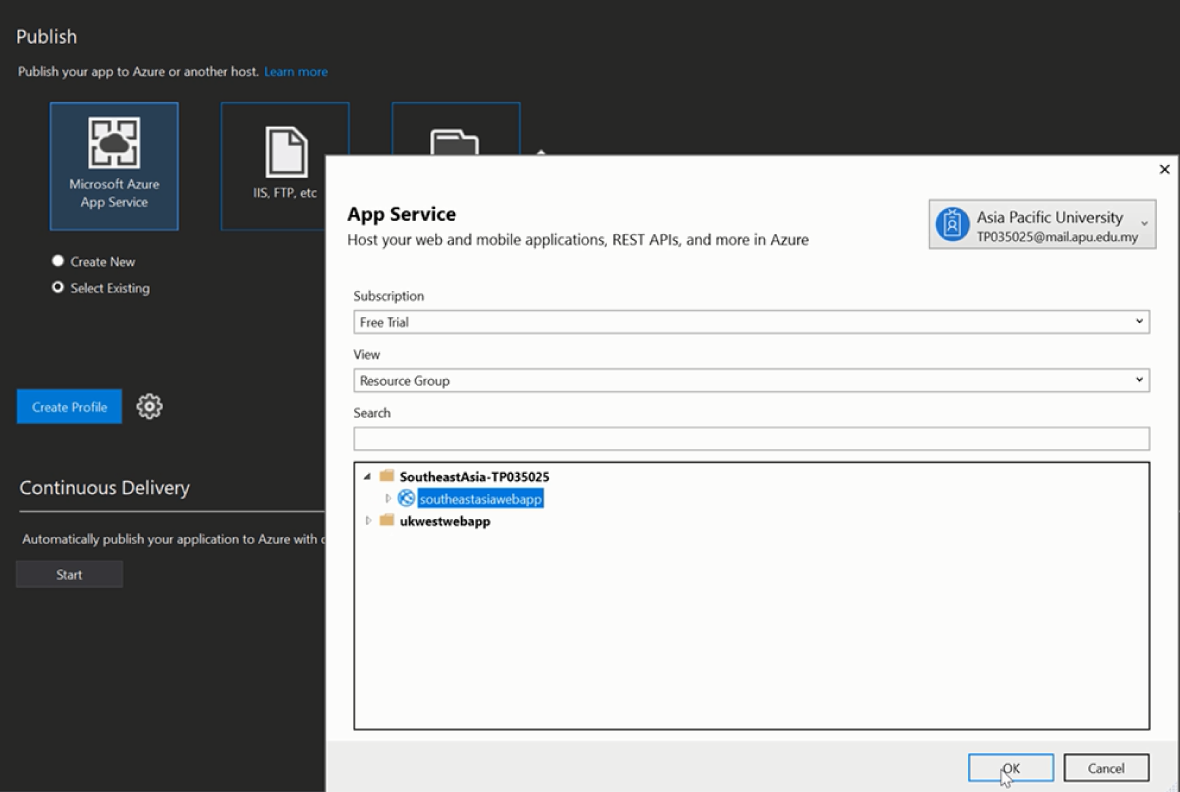


### 4.6.1 Endpoint set

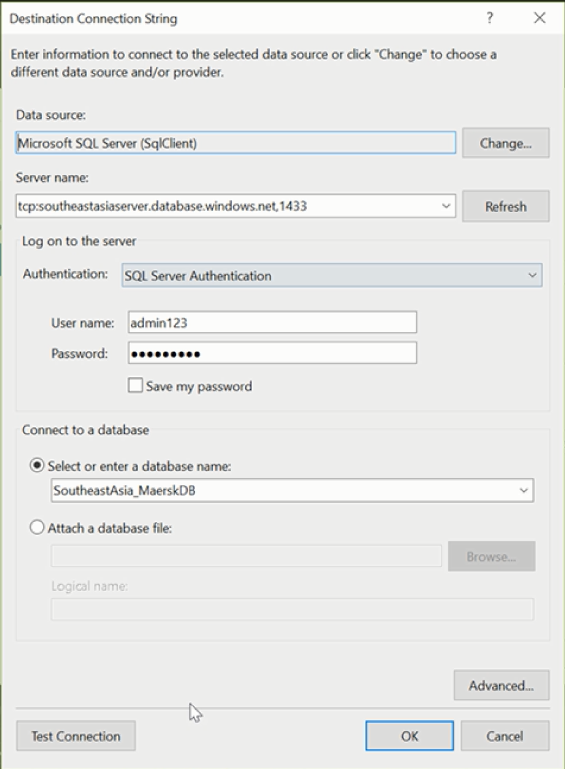


## 4.7 Deployment – Publish System to Online

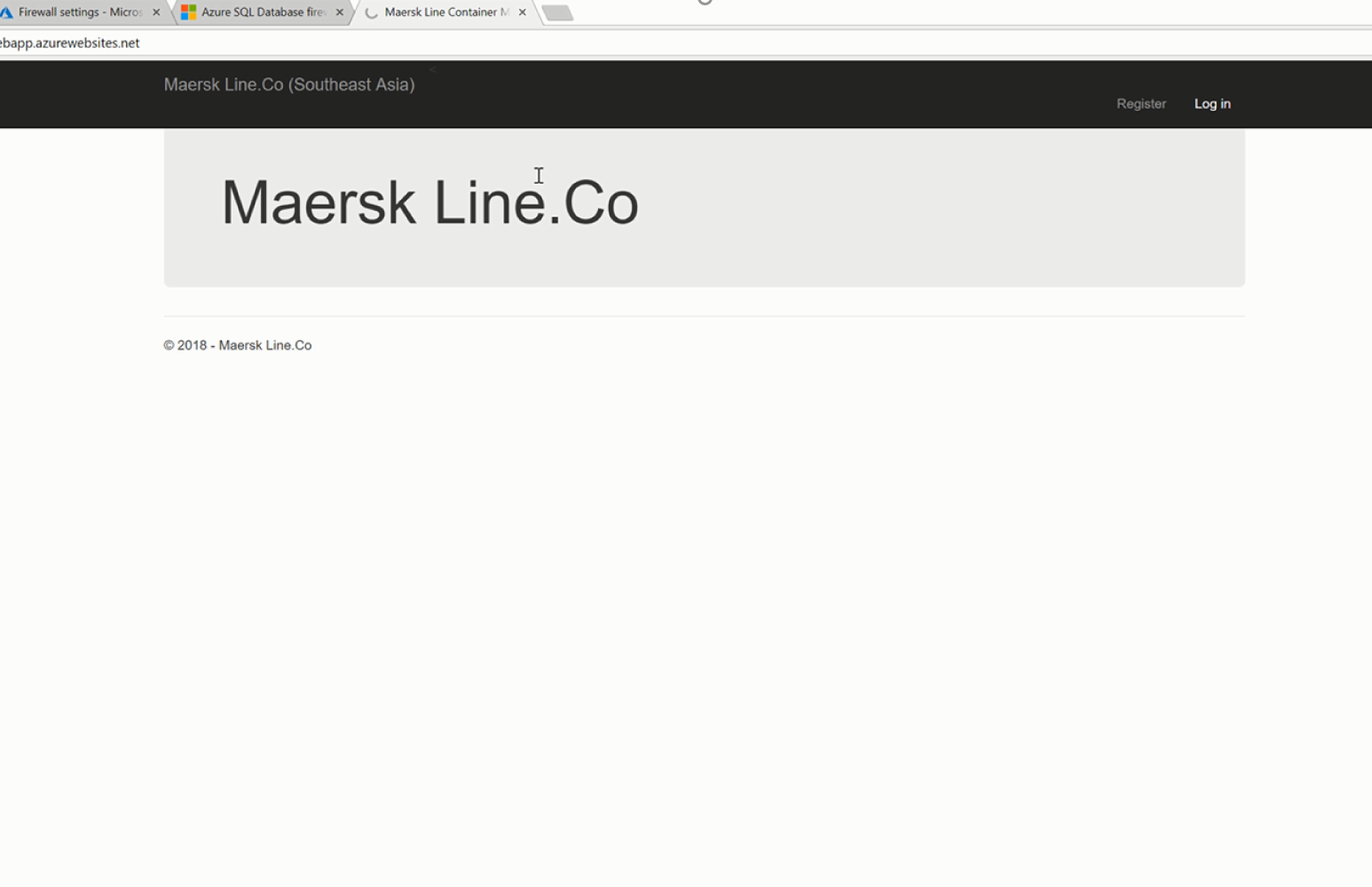
Using Visual Studio



### 4.7.1 Setting up Admin username and password



### 4.8 Published Successful – Southeast Asia



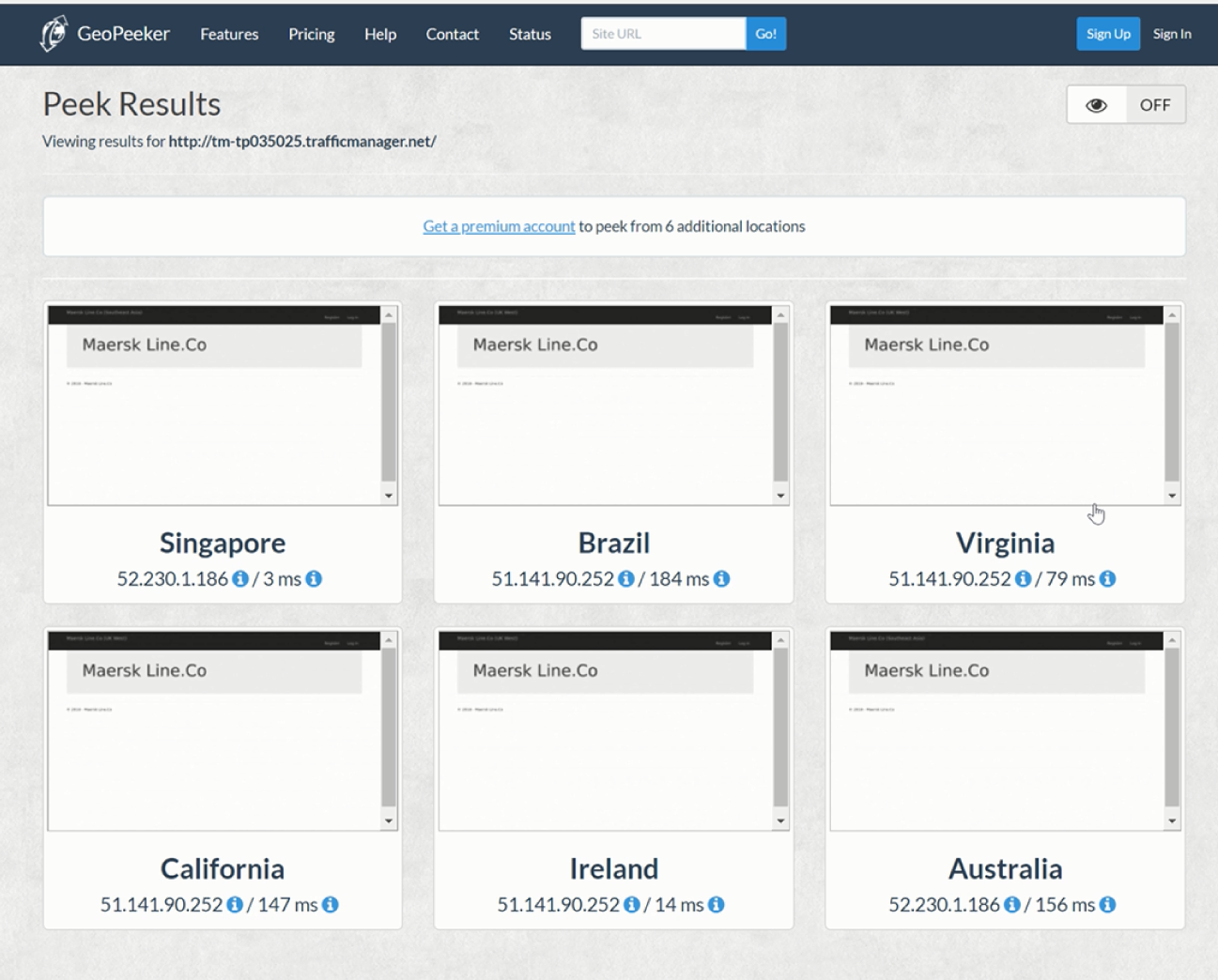
### 4.8.1 Published Successful – UK West



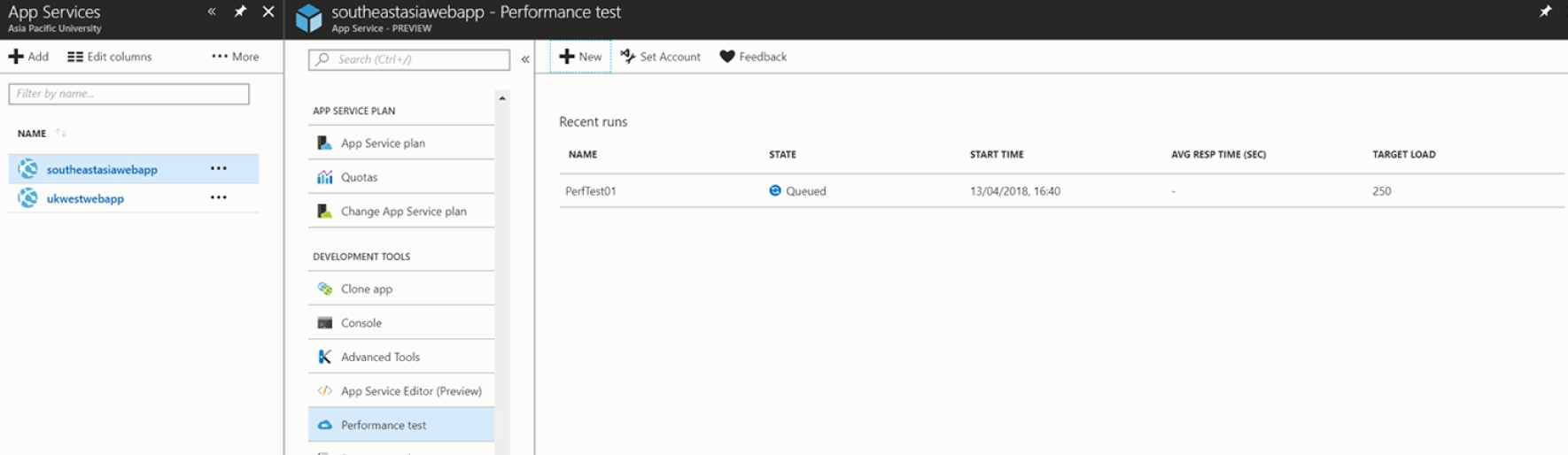
4.9 Testing Traffic Manager and Endpoints

After configuring traffic manager and endpoints, the URL is tested by accessing from different countries, using an online service call “GeoPeeker”.

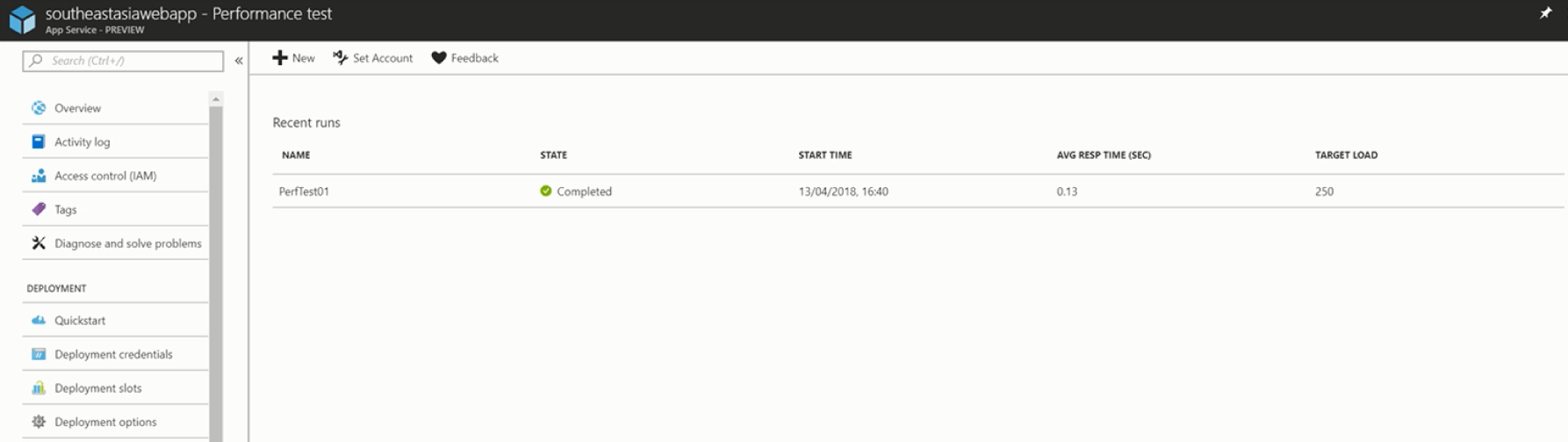
* Singapore and Australia using IP Address – 52.230.1.186 (Southeast Asia)
* Brzil, Virginia, California and Ireland – 51.141.90.252 (UK West)

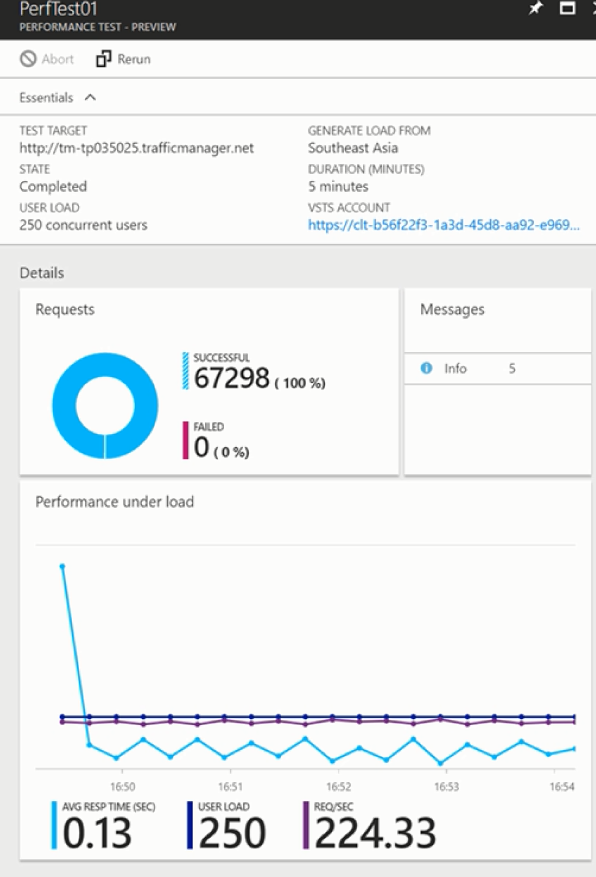


4.9.1Performance test for southeastasiawebapp server



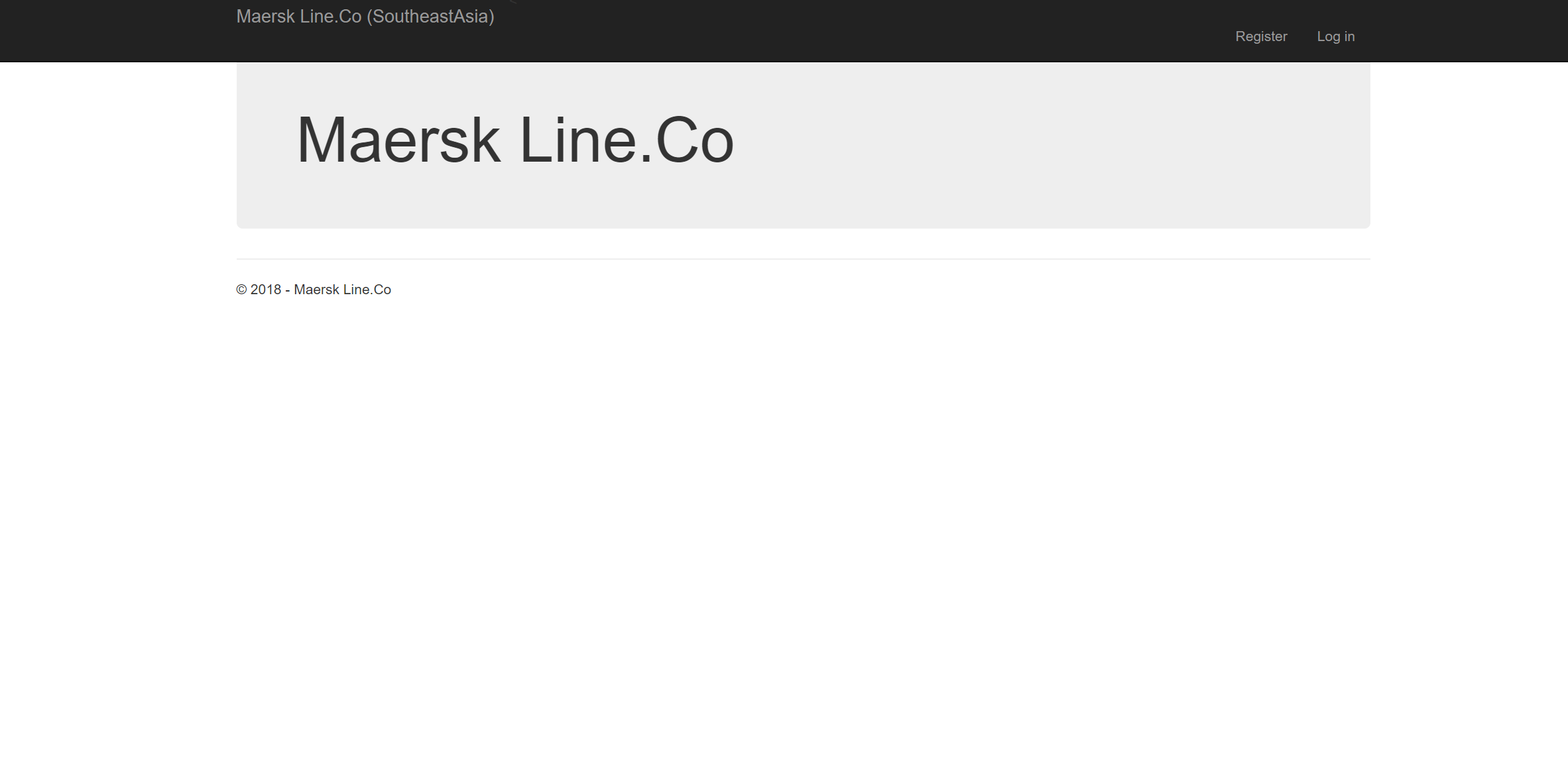
#### 4.9.1.1 Performance Test Result



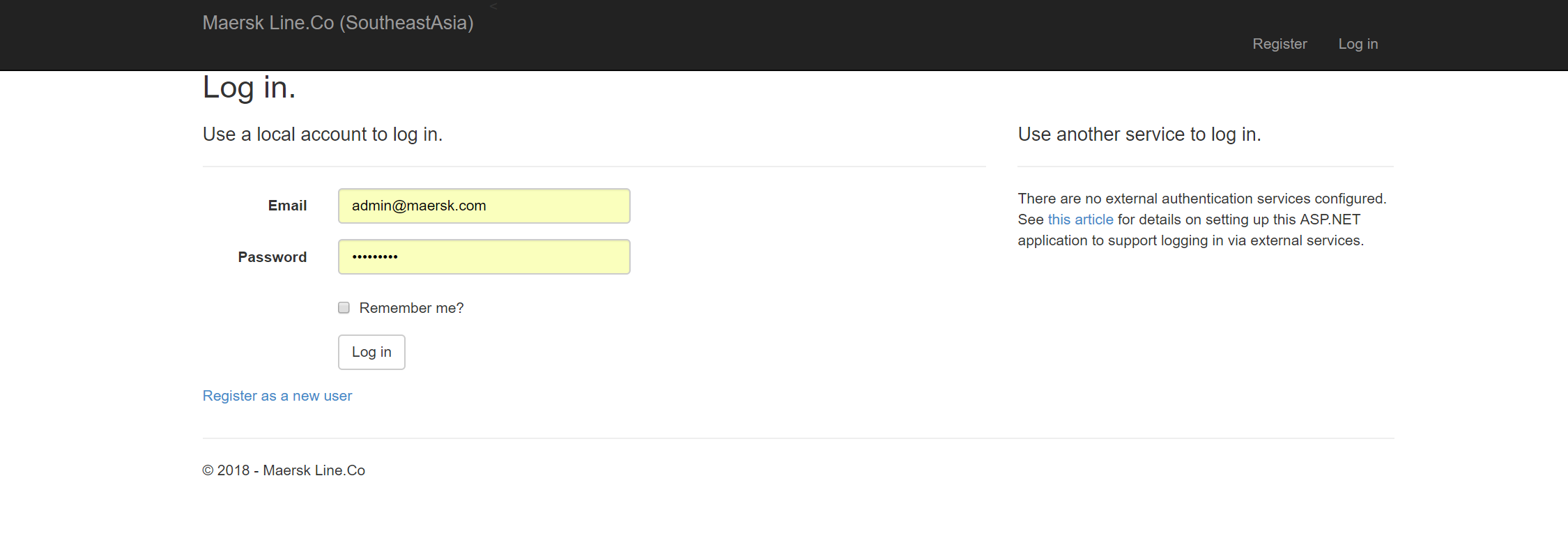


# 5.0 System Interfaces

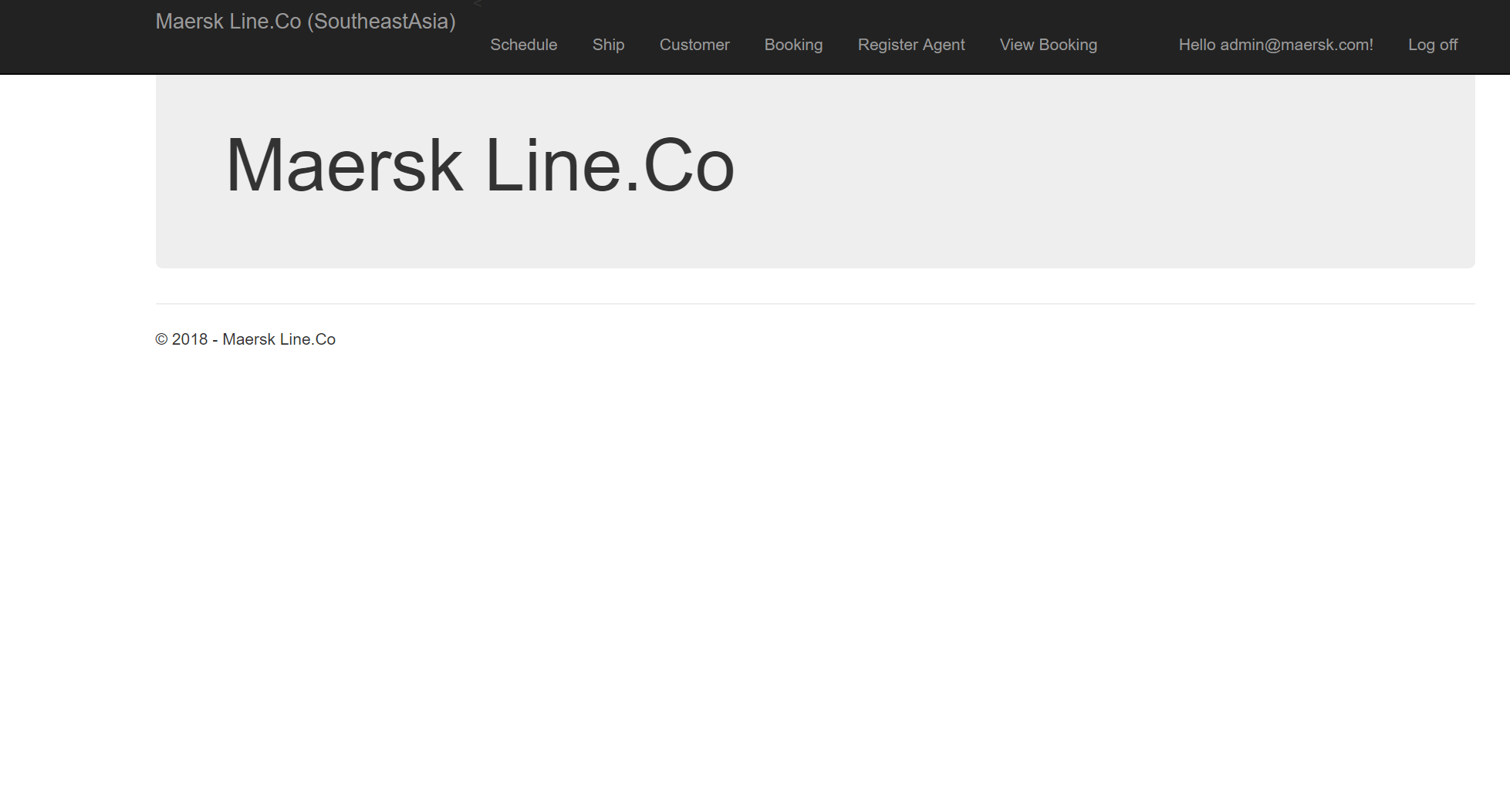
## 5.1 Main Page



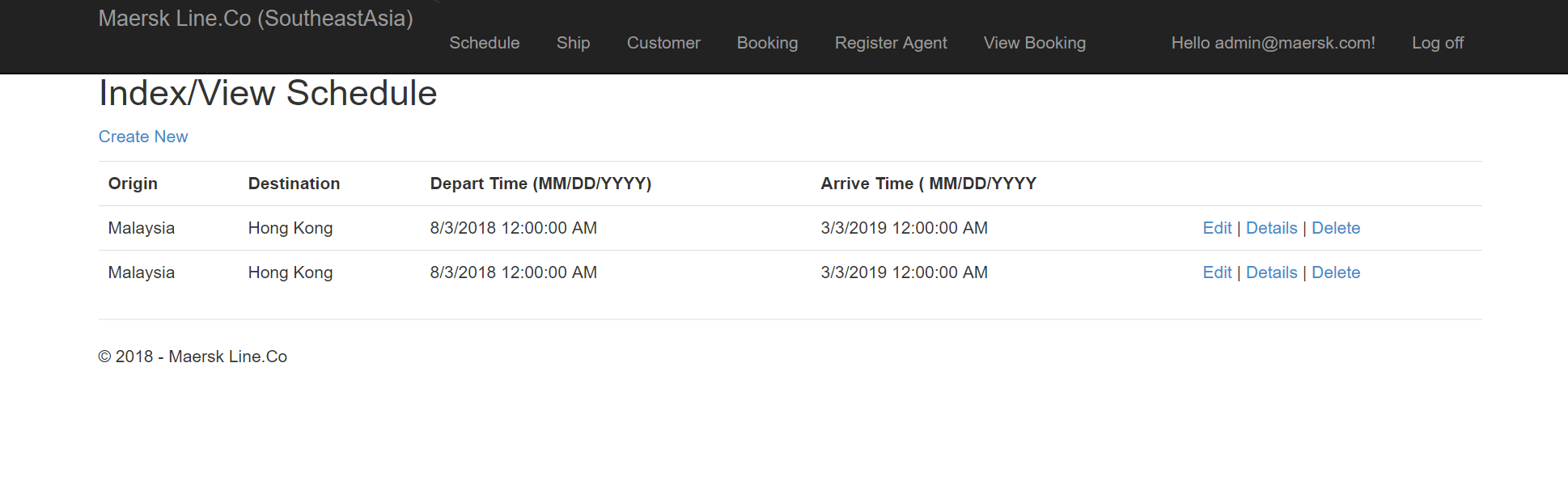
## 5.2 Login Page – Login as Admin



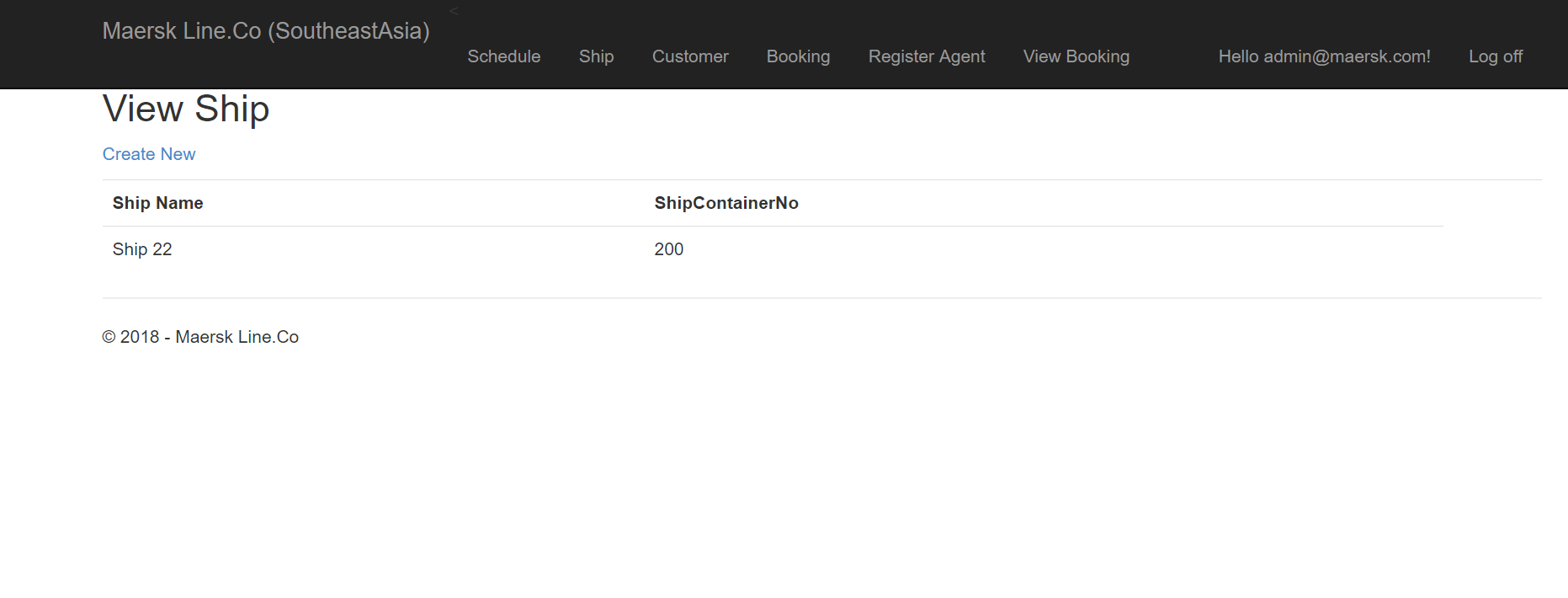
## 5.3 Home Page



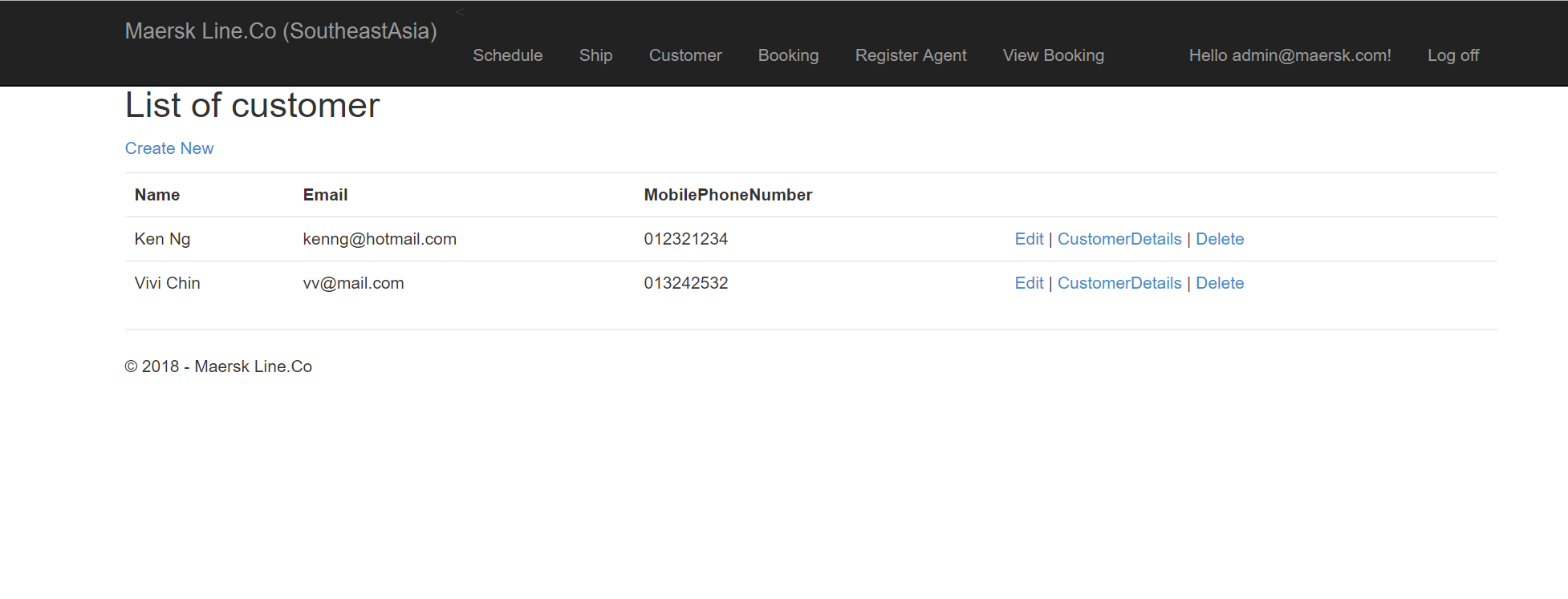
## 5.4 Schedule Page



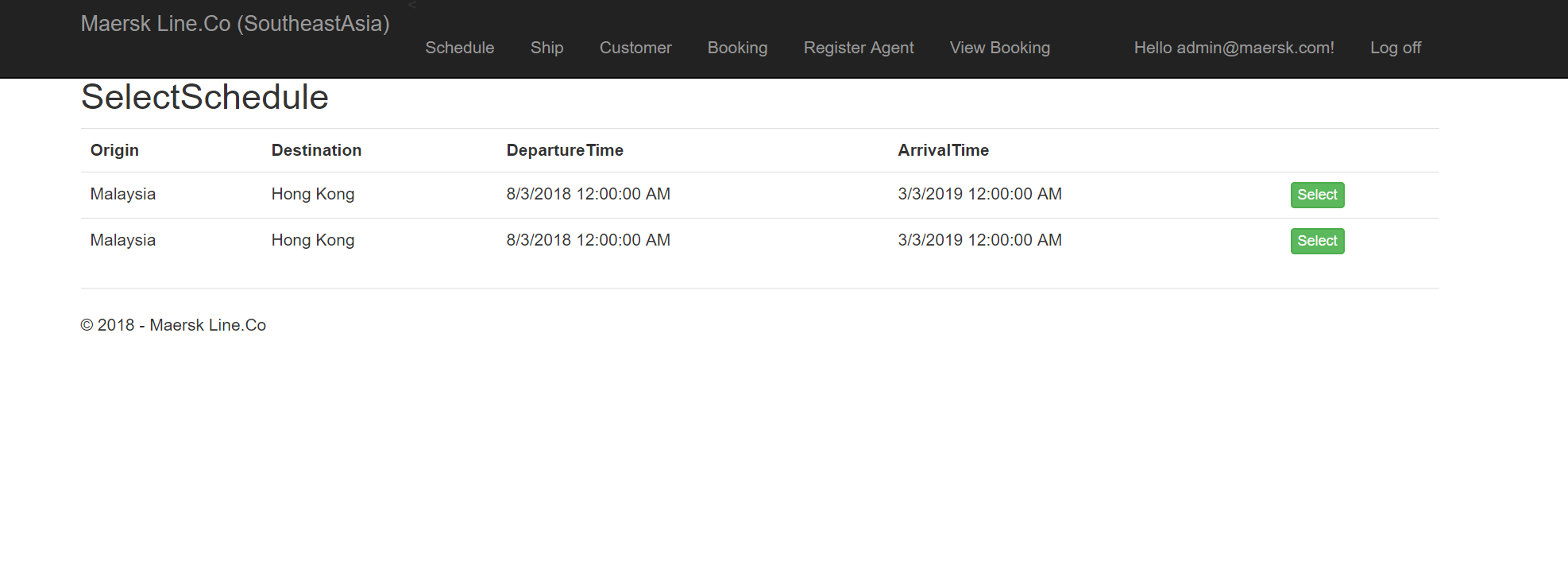
## 5.5 Ship Page



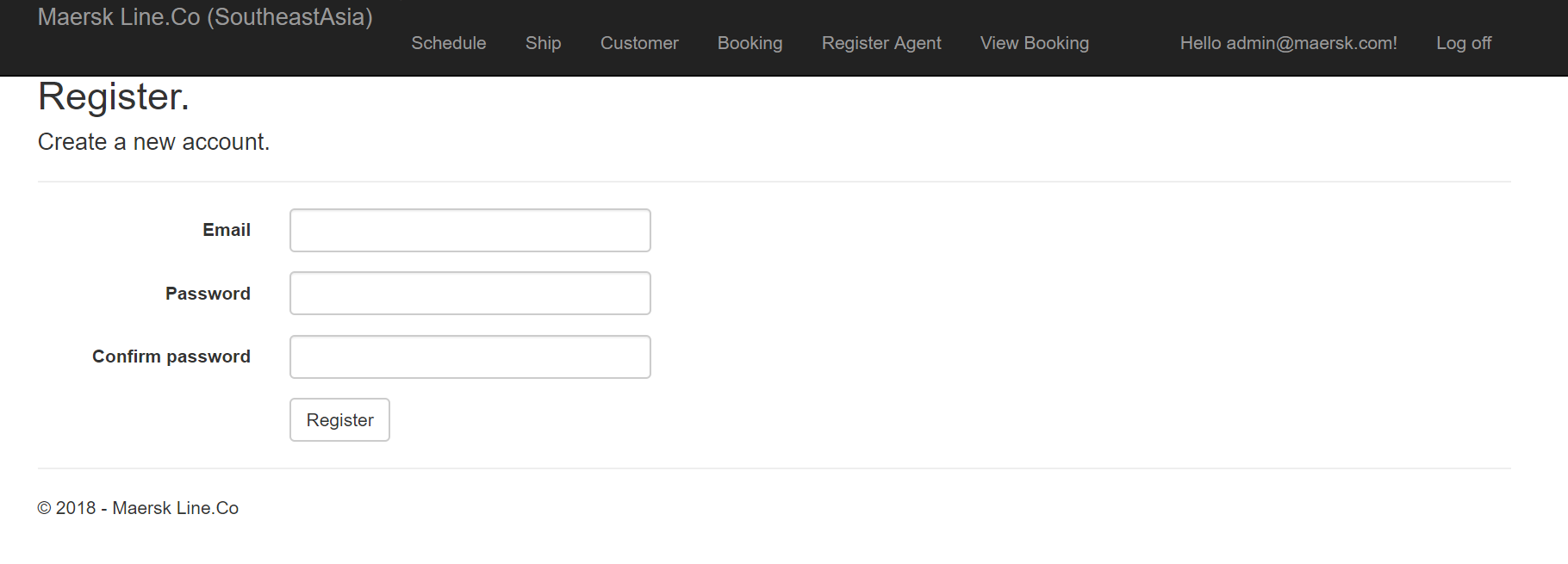
## 5.6 Customer Page



## 5.7 Booking Page



## 5.8 Register Agent Page



## 5.9 Unit Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | TEST CASE | TEST DATA | EXPECTED RESULT | ACTUAL RESULT | STATUS |
| T01 | Login | admin@maersk.com | Login Success | Login Success | Pass |
| T02 | Logout |  | Logout Success | Logout Success | Pas |
| T03 | Create Ship | Ship 22, 200 | Inserted | Inserted | Pass |
| T04 | Create Schedule | Malaysia, Hong Kong | Inserted | Inserted | Pass |
| T05 | Create Booking | Malaysia, Hong Kong | Inserted | Inserted | Pass |
| T06 | Create Customer | Ken Ng ,kenng@hotmail.com | Inserted | Inserted | Pass |

# 6.0 Conclusion

As conclusion, a highly scalable and efficient cloud-based application for Maersk Line company has been developed and published to Microsoft Azure cloud platform. The Maersk Line Company is now able to accommodate business growth and support its daily container management routines. By having this cloud-based application, the company also able to reduce their supply chain costs while efficiently manage improve the logistic activities for current and future used.

According to the cloud architecture diagram developed for Maersk Line Container Management, several types of Azure cloud tools has been utilised and used for the project development to ensure the application performance. Some of the services being used include Azure web application service, SQL server, SQL database, Geo-replication, auto, traffic manager, and performance test.

# Reference

1) Microsoft Azure, 2017. Autoscaling. Microsoft Azure. [ONLINE] Available at: https://docs.microsoft.com/en-us/azure/architecture/best-practices/auto-scaling [Accessed on: 10 March 2017]

2) Microsoft, 2017. 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-serveriaas [Accessed on: 11 March 2017]

# Appendix

Account Username and Password

Admin :

Username: [admin@maersk.com](mailto:admin@maersk.com)

Password : Admin123.

Agent:

Username: [agent@maersk.com](mailto:agent@maersk.com)

Password : Agent123.

Published Online

* WebLink : <http://tm-tp035025.trafficmanager.net/> (southeast asia)
* Weblink : <http://ukwestwebapp.azurewebsites.net/> (uk west)

Username: admin123

Password : south123.

Github Link

<https://github.com/lam96/DDAC-TP035025>