INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

Designing & Developing Cloud Applications

CT071-3-5-3-DDAC

Name	**:**	YAP YEE XIANG
Student ID	**:**	TP037636
Intake Code	**:**	UC3F1706 SE
Lecturer Name	**:**	DR KALAI ANAND RATNAM
Hand out Date	**:**	6th December 2017
Hand in Date	**:**	13th April 2018

ACKNOWLEDGEMENT

Firstly, I would like to thank my parents for their continuous support in my studies. I would also like to thank my lecturer Dr. Kalai for his guidance and expertise in the field of cloud computing especially in using the Microsoft Azure. I would also like to thank my friends and fellow classmates for providing feedback to my ideas that would allow me to maintain the right path. My gratitude is also extended to my parents and they are accredited with highest respect for continuously support regardless of any situation.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 PROJECT BACKGROUND

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

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

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

In an effort to support further business growth and increase organizational flexibility, Maersk decided to consolidate all of its data 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.

1.2 OBJECTIVE & SCOPE

- To provide the fundamental understanding of application in cloud computing in its different forms and how Microsoft Azure fits within the cloud computing space.
- To explore, experiment and experience the Microsoft Azure development environment.
- Design, implementation and deployment of the Maersk Line Container Management System (CMS) to the Microsoft Azure.
- To architecturally design efficient Maersk Line CMS by using Microsoft Azure as the public cloud platform.

1.3 REQUIREMENT SPECIFICATIONS

The developer is required to design and develop a single tenant web solution that meets the following criteria: -

1. From import, export and transhipment processing to gate operations.
2. To be able to scale the solution to meet the needs of demands during peak seasons.
3. Improves profitability, reduce costs, increases productivity, eradicates errors and optimizes resources to future-proof your cargo handling business for high performance.
4. Assurance & reliability through Failover Management.
5. 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.
6. Manage your entire booking process from schedule search to booking confirmation.

1.4 SUMMARY OF MAJOR FUNCTION/SOLUTION CONTENTS

1. Design and develop a single tenant web application hosted on Microsoft Azure as an App Service (Web App).
2. Consume Azure Storage or SQL Database.
3. Consist of 5 – 10 interlinked pages.
4. Provide quality content and design.
5. Analyse web application performance with monitoring tools.
6. To be able to scale the solution to meet the needs of demands during peak seasons.
7. Source code to place in source control management services.

2. PROJECT PLAN

ID	Task Name	Duration	Start	Finish	Predecessors
1	**Maersk Line Container Management System**	**77 days**	**Mon 8/1/18**	**Tue 24/4/18**	
2	**Introduction**	**5 days**	**Mon 8/1/18**	**Fri 12/1/18**	
3	Project Background	1 day	Mon 8/1/18	Mon 8/1/18	
4	Objective	1 day	Tue 9/1/18	Tue 9/1/18	3
5	Scope	1 day	Wed 10/1/1	Wed 10/1/1	4
6	Requirement Specifications	1 day	Thu 11/1/18	Thu 11/1/185	
7	Summary of Solution	1 day	Fri 12/1/18	Fri 12/1/18	6
8	Project Introduction Completed	0 days	Fri 12/1/18	Fri 12/1/18	7
9	**Analysis**	**9 days**	**Mon 15/1/1**	**Thu 25/1/1i2**	
10	Analyze Business Needs	3 days	Mon 15/1/18	Wed 17/1/18	8
11	Analyze Requirement Specifications	3 days	Thu 18/1/18	Mon 22/1/18	10
12	Analyze Deliverables and Fulfillment	3 days	Tue 23/1/18	Thu 25/1/18	11
13	Project Anaylsis Completed	0 days	Thu 25/1/18	Thu 25/1/1812	
14	**Design**	**11 days**	**Fri 26/1/18**	**Fri 9/2/18**	
15	Design Cloud Design Patterns	3 days	Fri 26/1/18	Tue 30/1/1813	
16	Design Architectural Diagrams	3 days	Wed 31/1/18	Fri 2/2/18	15
17	Design Considerations	3 days	Mon 5/2/18	Wed 7/2/1816	
18	Design Modelling	2 days	Thu 8/2/18	Fri 9/2/18	17
19	Web Application Design Completed	0 days	Fri 9/2/18	Fri 9/2/18	18
20	**Implementation**	**30 days**	**Tue 13/2/18**	**Mon 26/3/1**	
21	Web Application Development	28 days	Tue 13/2/18	Thu 22/3/1819	
22	Azure Service Integration	1 day	Fri 23/3/18	Fri 23/3/18	21
23	Web Application Publish with Azure	1 day	Mon 26/3/18	Mon 26/3/18	22
24	Web Applilication Implementation Completed	0 days	Mon 26/3/18	Mon 26/3/18	23
25	**Test Plan & Testing Discussion**	**13 days**	**Tue 27/3/18**	**Thu 12/4/1i20**	
26	Test Plan Development	3 days	Tue 27/3/18	Thu 29/3/18	
27	Conduct Unit Testing	5 days	Fri 30/3/18	Thu 5/4/18	26
28	Conduct Performance Testing	5 days	Fri 6/4/18	Thu 12/4/1827	
29	Test Plan Execution & Testing Discussion Completed	0 days	Thu 12/4/18	Thu 12/4/18	28
30	**Project Final Documentation**	7 days	Fri 13/4/18	Mon 23/4/18	29
31	**Conclusion**	1 day	Tue 24/4/18	Tue 24/4/1830	
32	**Maersk Line Container Management System Completed**	0 days	Tue 24/4/18	Tue 24/4/18	31

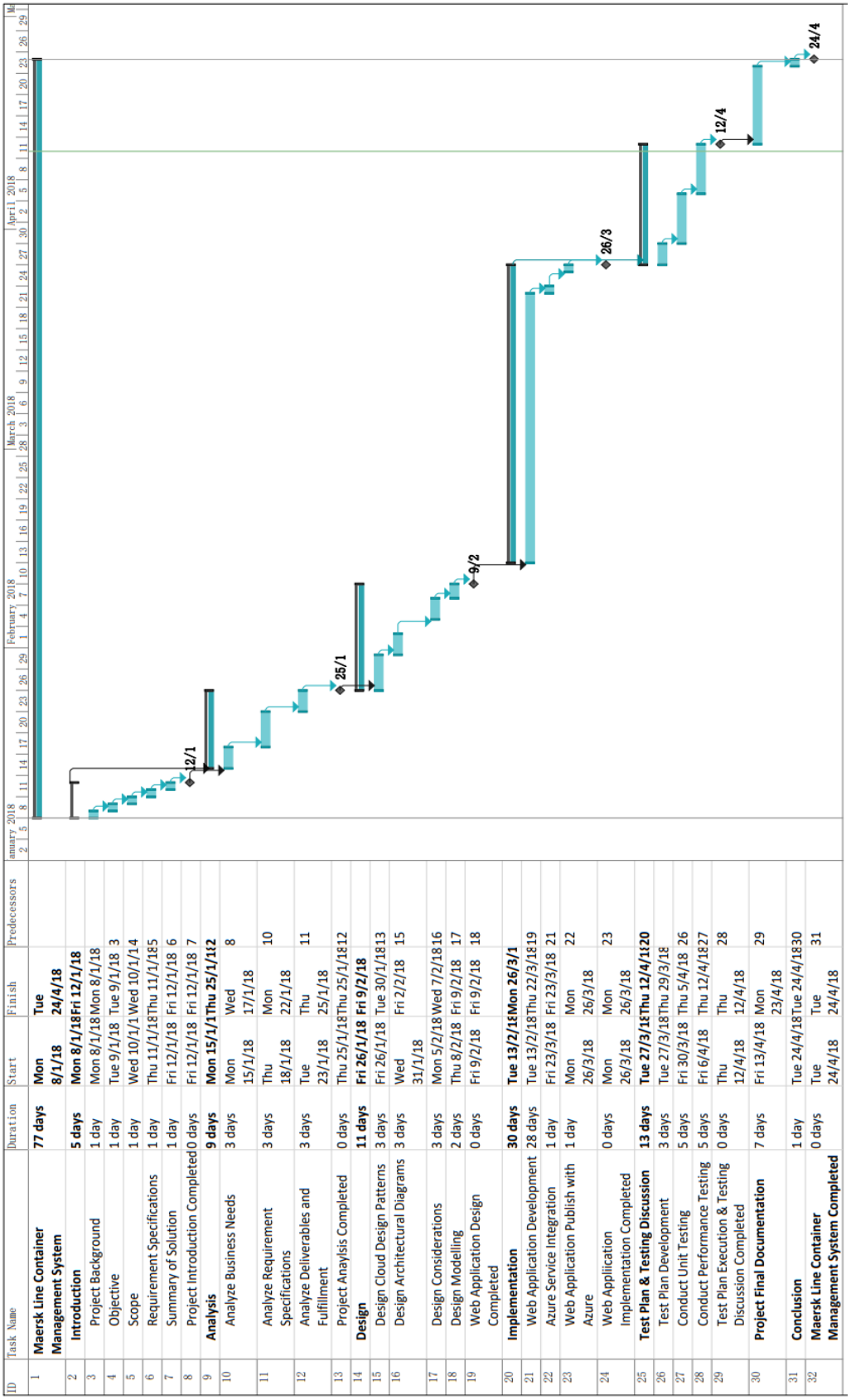


Figure 2-1 - Gantt Chart of the Project Lifetime

3. DESIGN

3.1 DESIGN CONSIDERATION

There are some assumption or considerations that have been provided throughout the project lifetime such as listed below: -

1. The number of container can be booked in each shipment booking is limited to 1.
2. The development is conducted and focused on the Southeast Asia region.

3.2 MODELLING

3.2.1 Use Case Diagram

3.2.1.1 Maersk Line CMS – Agent Module Use Case Diagram

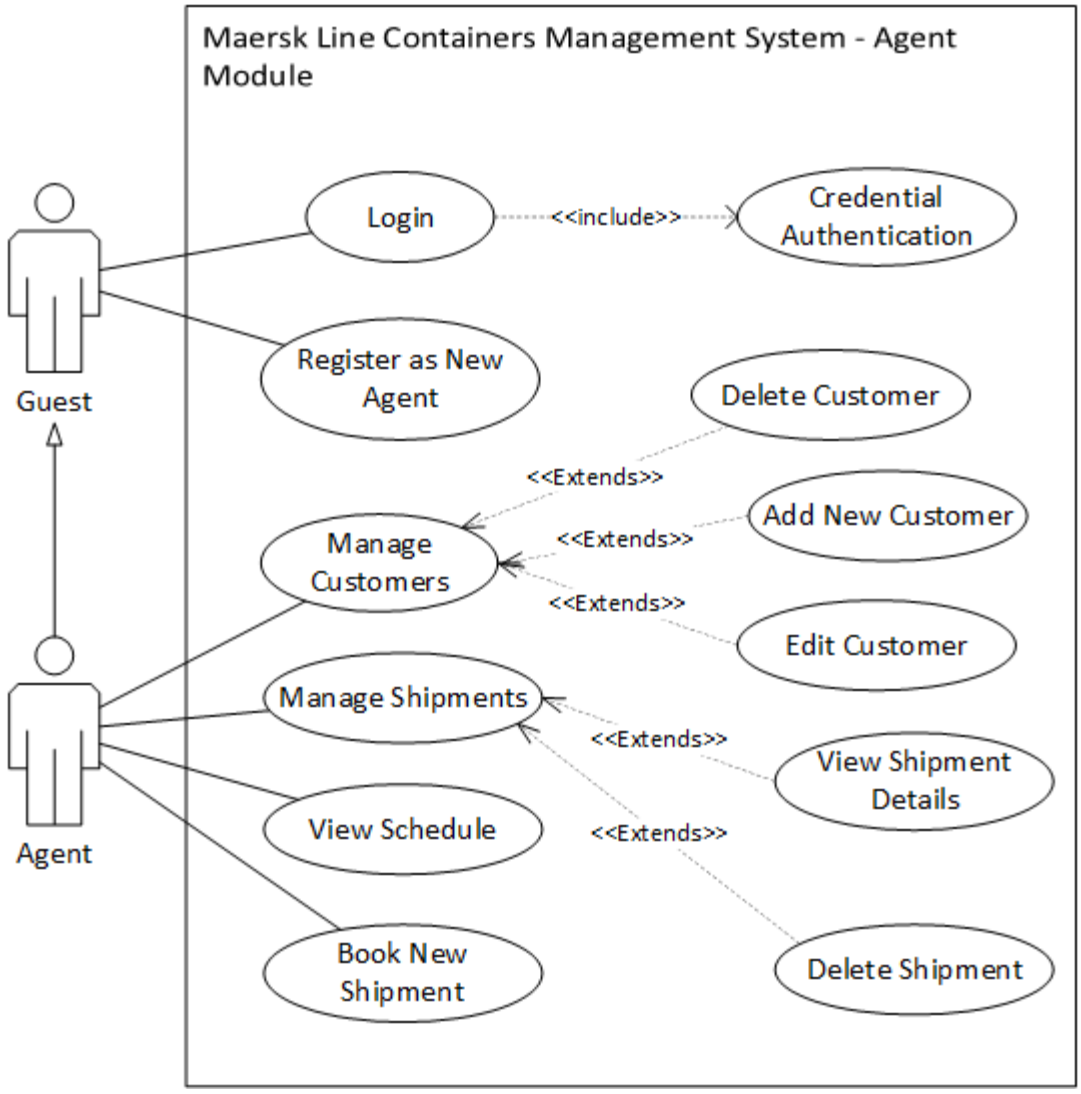


Figure 3-1 - Maersk Line CMS - Agent Module Use Case Diagram

3.2.1.2 Maersk Line CMS – Administrator Module Use Case Diagram

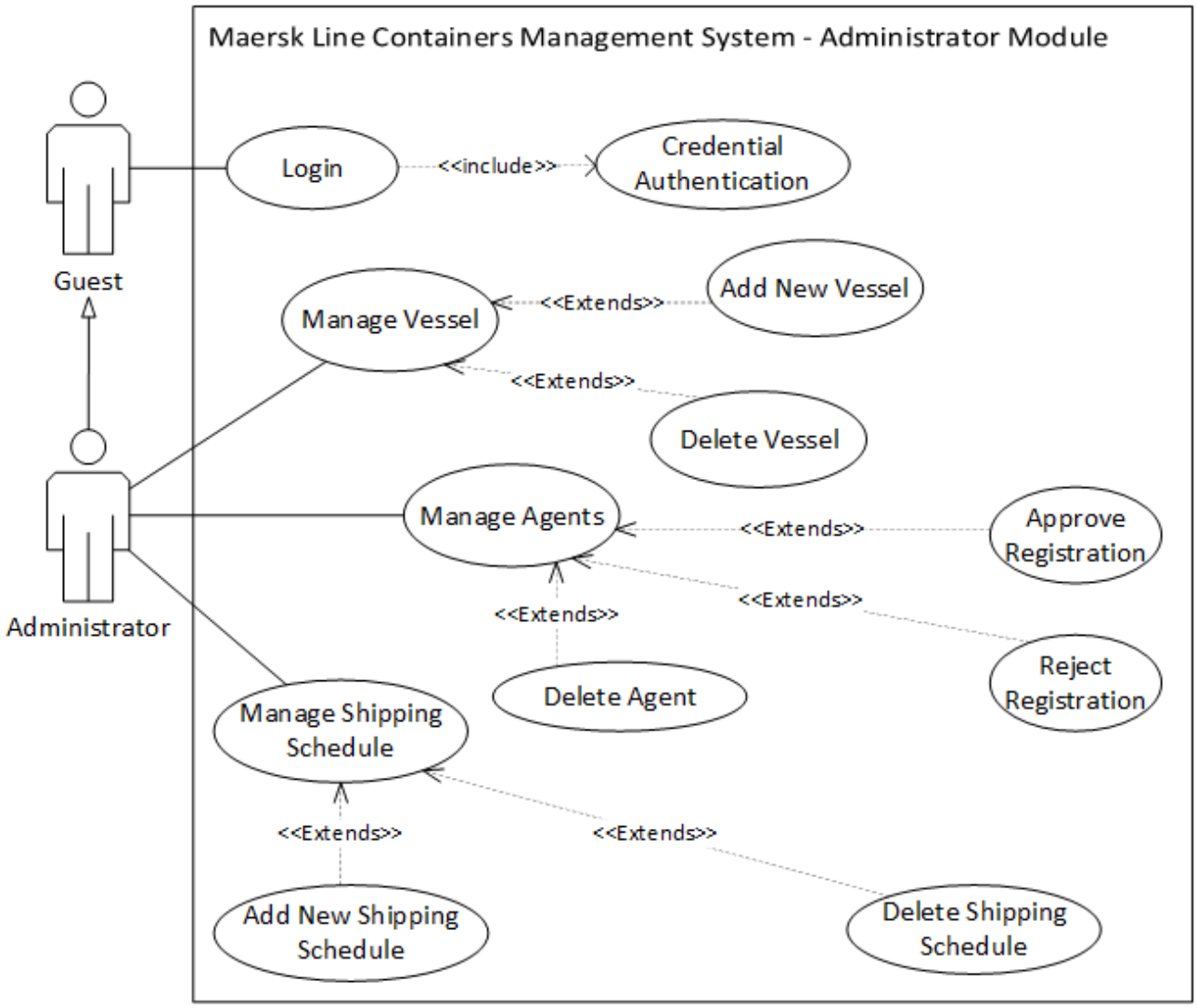


Figure 3-2 - Maersk Line CMS - Administrator Module Use Case Diagram

3.2.2 Use Case Description

3.2.2.1 Maersk Line CMS – Agent Module Use Case

3.2.2.1.1 Login

Use Case:	Login
Summary:	User enter the username and password to request appropriate access to the system.
Dependency:	<<Include>> Credential Authentication
Actors:	Guest
Precondition(s):	User has connected and opened the Agent login page.
Main Sequence:	1. User provides username and password and press login button. 2. System capture user's input and verify the authentication from web server. If username and password found in web server, the web server confirms the credential and grants appropriate access to user as Agent.
Alternative Sequence:	**Step 2:** If username and password not found in account database, the system displays an error message to user and prompt user for next attempt.
Postcondition(s):	Web Server has identified the account type and redirected to the appropriate page with appropriate user access.

Table 3-1 - Login Use Case Specification

3.2.2.1.2 Register as New Agent

Use Case:	Register as New Agent
Summary:	User register a new agent account before using the service.
Dependency:	-
Actors:	Guest
Precondition(s):	User has connected and opened the Register page.
Main Sequence:	1. User enter the required information and pressed the Register button. 2. The browser will submit the registration info to the web server. 3. The web server validates the entered information. 4. If the username is not existed, the web server will respond to the browser to redirect the user to the success page.
Alternative Sequence:	**Step 2:** If username exists account database, the system displays an error message to user and prompt user for next attempt.
Postcondition(s):	User has been redirected to the success page.

Table 3-2 - Register as New Agent Use Case Specification

3.2.2.1.3 Manage Customers

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.

Table 3-3 - Manage Customers Use Case Specification

3.2.2.1.4 Delete Customer

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.

3.2.2.1.5 Add New Customer

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.

3.2.2.1.6 Edit Customer

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.

3.2.2.1.7 Manage Shipments

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.

3.2.2.1.8 View Shipment Details

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.

3.2.2.1.9 Delete Shipment

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:	3. User has clicked on Delete link from a selected shipment record. 3. 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.

3.2.2.1.10 View Schedule

Use Case:	View Schedule
Summary:	Agent view the available shipping schedules
Dependency:	-
Actors:	Agent
Precondition(s):	User has logged in as Agent and click on Shipping Schedule 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:	-
Postcondition(s):	User has been redirected to the Booking Page.

3.2.2.1.11 Book New Shipment

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.

3.2.2.2 Maersk Line CMS – Administrator Module Use Case

3.2.2.2.1 Login

Use Case:	Login
Summary:	User enter the username and password to request appropriate access to the system.
Dependency:	<<Include>> Credential Authentication
Actors:	Guest
Precondition(s):	User has connected and opened the Admin login page.
Main Sequence:	3. User provides username and password and press login button. 4. System capture user's input and verify the authentication from web server. If username and password found in web server, the web server confirms the credential and grants appropriate access to user as Agent.
Alternative Sequence:	**Step 2:** If username and password not found in account database, the system displays an error message to user and prompt user for next attempt.
Postcondition(s):	Web Server has identified the account type and redirected to the appropriate page with appropriate user access.

3.2.2.2.2 Manage Vessel

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.

3.2.2.2.3 Create New Vessel

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.

3.2.2.2.4 Delete Vessel

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.

3.2.2.2.5 Manage Agents

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.

3.2.2.2.6 Delete Agent

Use Case:	Delete Agent
Summary:	Admin delete the selected existing agent 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 Delete link from a selected agent record in the Agent List table. 2. The web server will delete the selected agent record from database and reload the page.
Alternative Sequence:	-
Postcondition(s):	The selected agent record has been removed from the database and the Manage Agents page has been reloaded.

3.2.2.2.7 Approve Registration

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.

3.2.2.2.8 Reject Registration

Use Case:	Reject Registration
Summary:	Admin reject 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 Reject link from a selected agent record in the New Registration to be Approved table. 2. The web server will delete the selected agent record from database and reload the page.
Alternative Sequence:	-
Postcondition(s):	The selected agent record has been removed from the database and the Manage Agents page has been reloaded.

3.2.2.2.9 Manage Shipping Schedule

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.

3.2.2.2.10 Add New Shipping Schedule

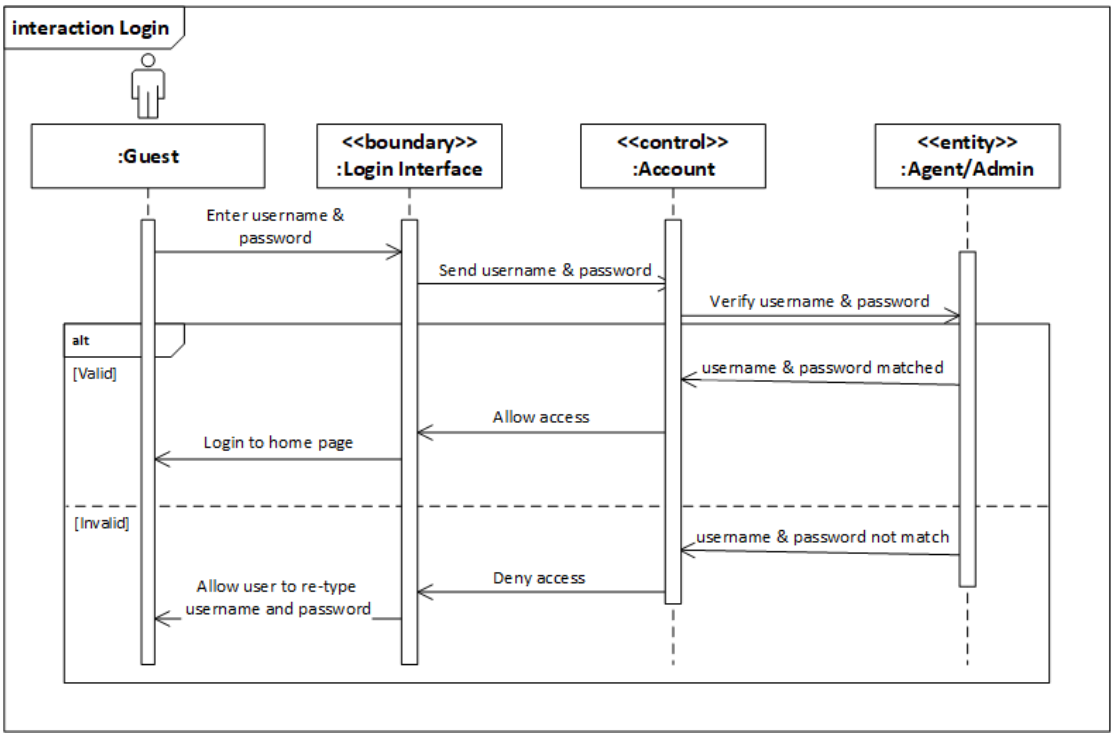
Use Case:	Add New Shipping Schedule
Summary:	Admin create a new shipping schedule record.
Dependency:	<<extends>> Manage Shipping Schedule
Actors:	Admin
Precondition(s):	User has clicked on Add New Shipping Schedule link in the Manage Shipping Schedule Page and redirected to the Add New Schedule Page.
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.

3.2.2.2.11 Delete Shipping Schedule

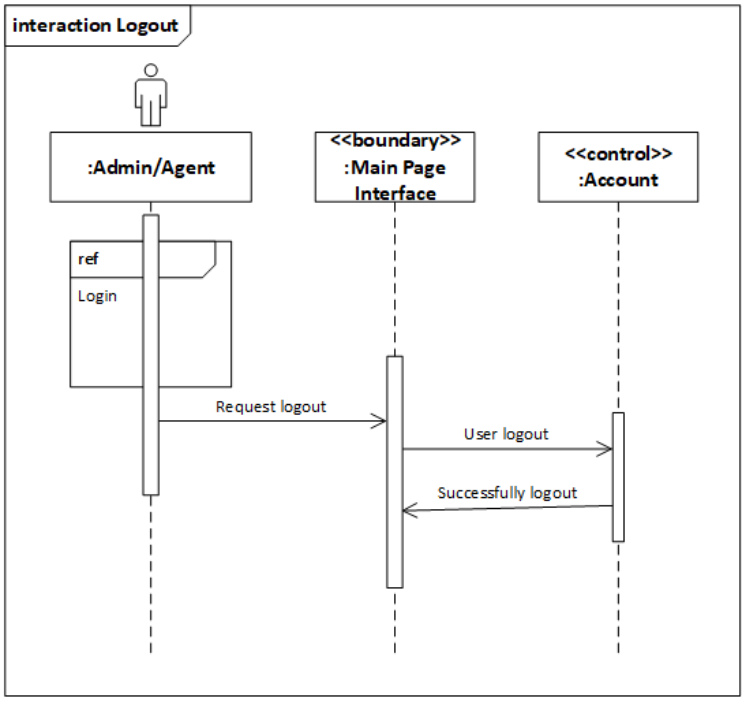
Use Case:	Delete Shipping Schedule
Summary:	Admin delete the selected existing agent record.
Dependency:	<<extends>> Manage Shipping Schedule
Actors:	Admin
Precondition(s):	User has logged in as Admin and click on Manage Shipping Schedule on Navigation Bar
Main Sequence:	1. User has clicked on Delete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

3.2.3 Sequence Diagram

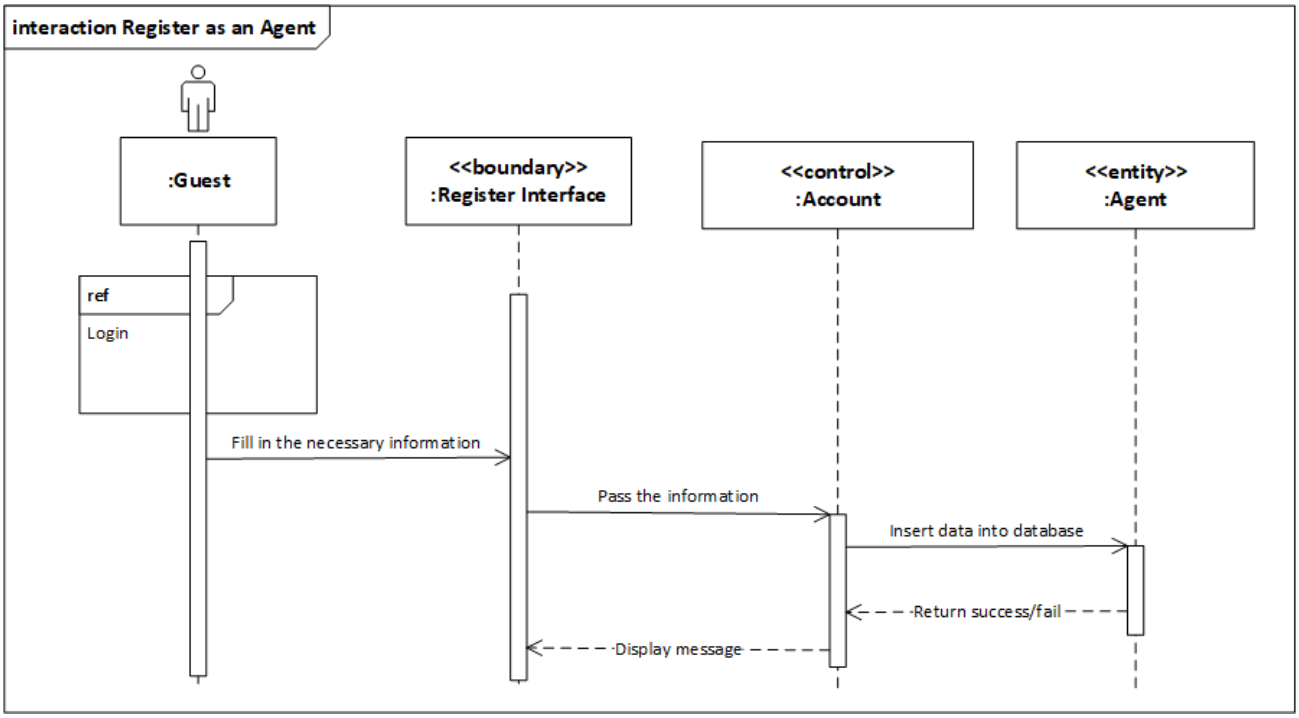
3.2.3.1 Login



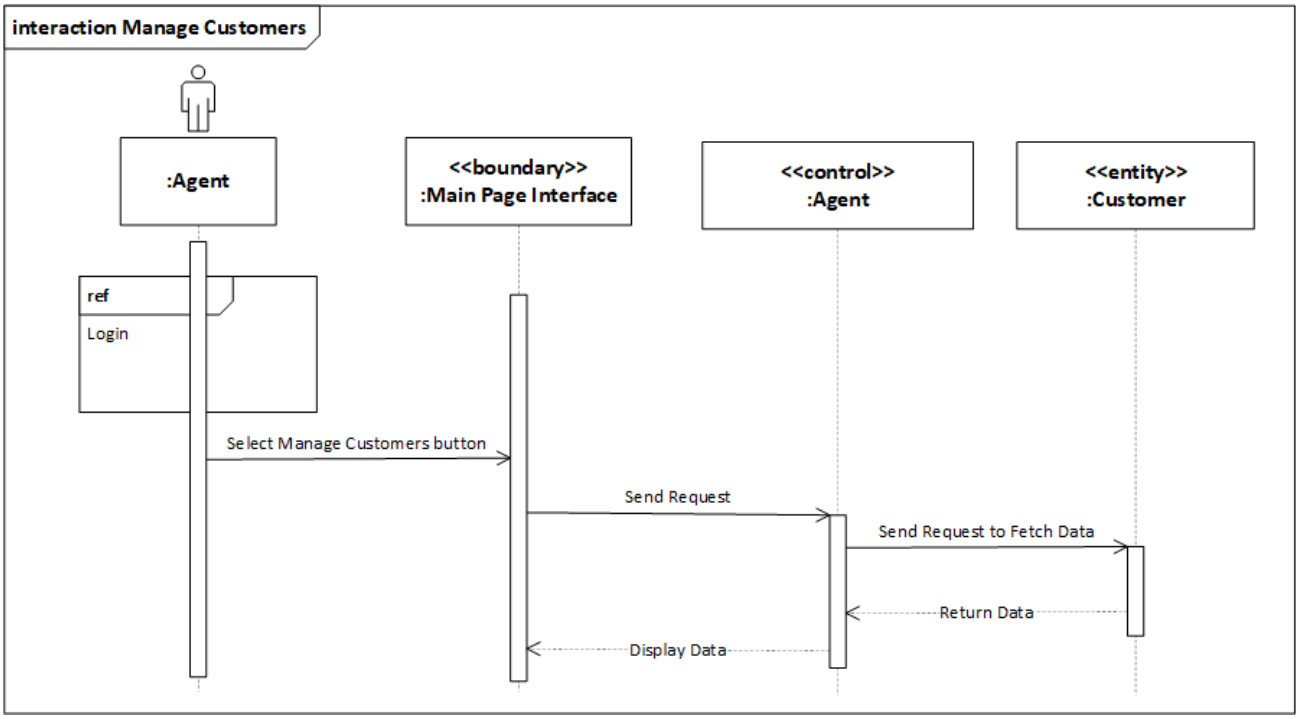
3.2.3.2 Logout



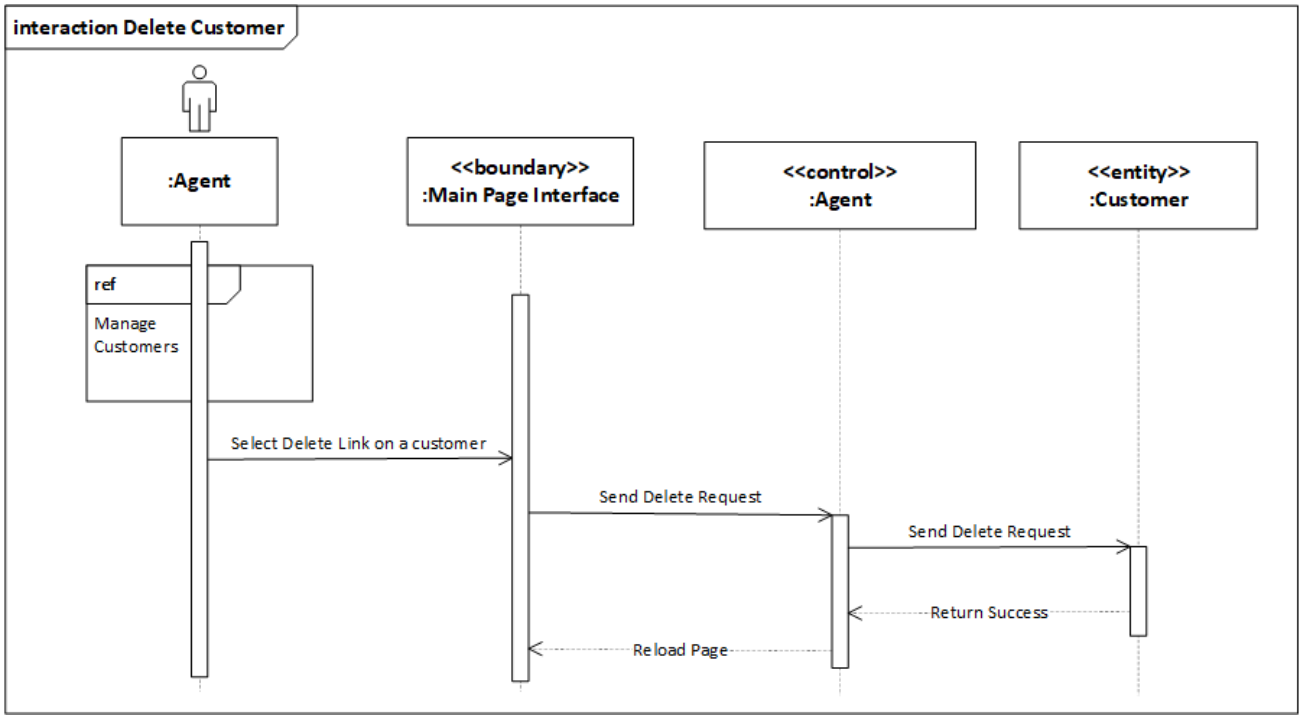
3.2.3.3 Register as an Agent



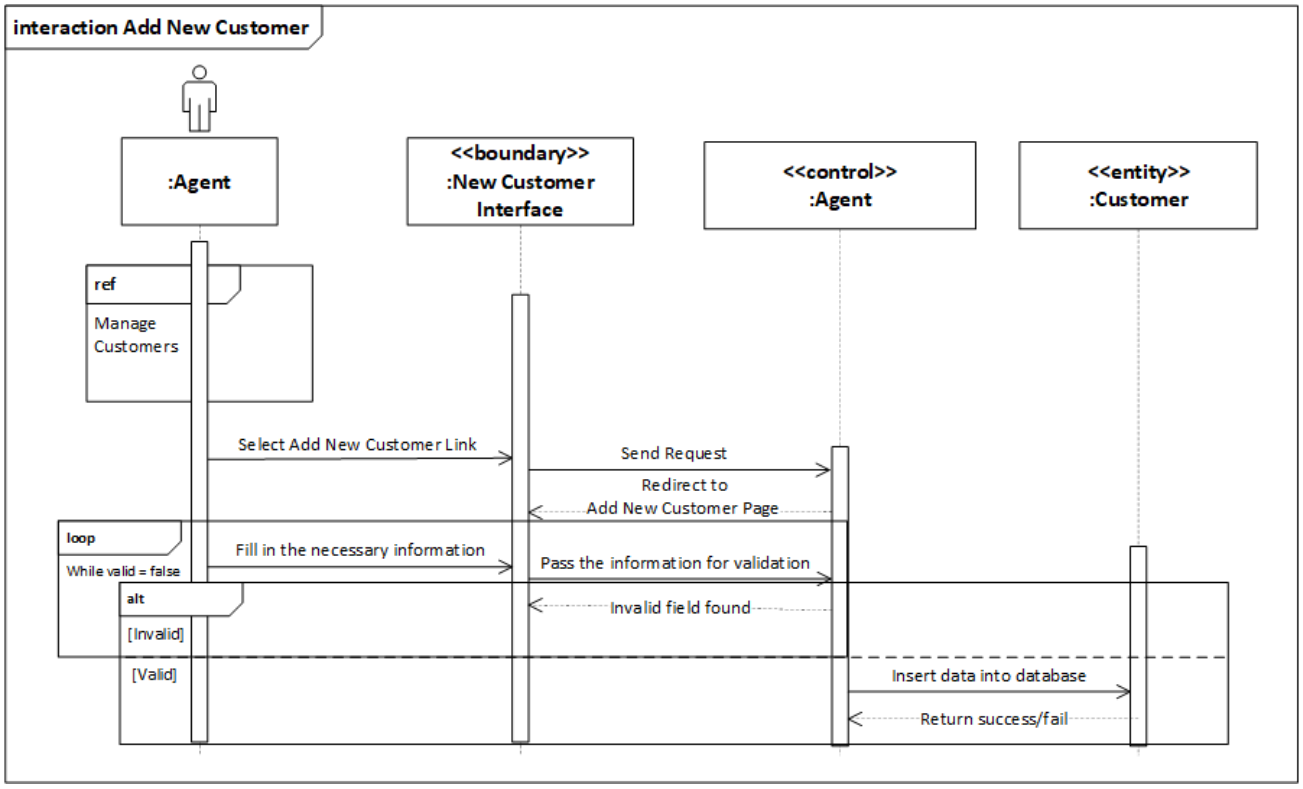
3.2.3.4 Manage Customers



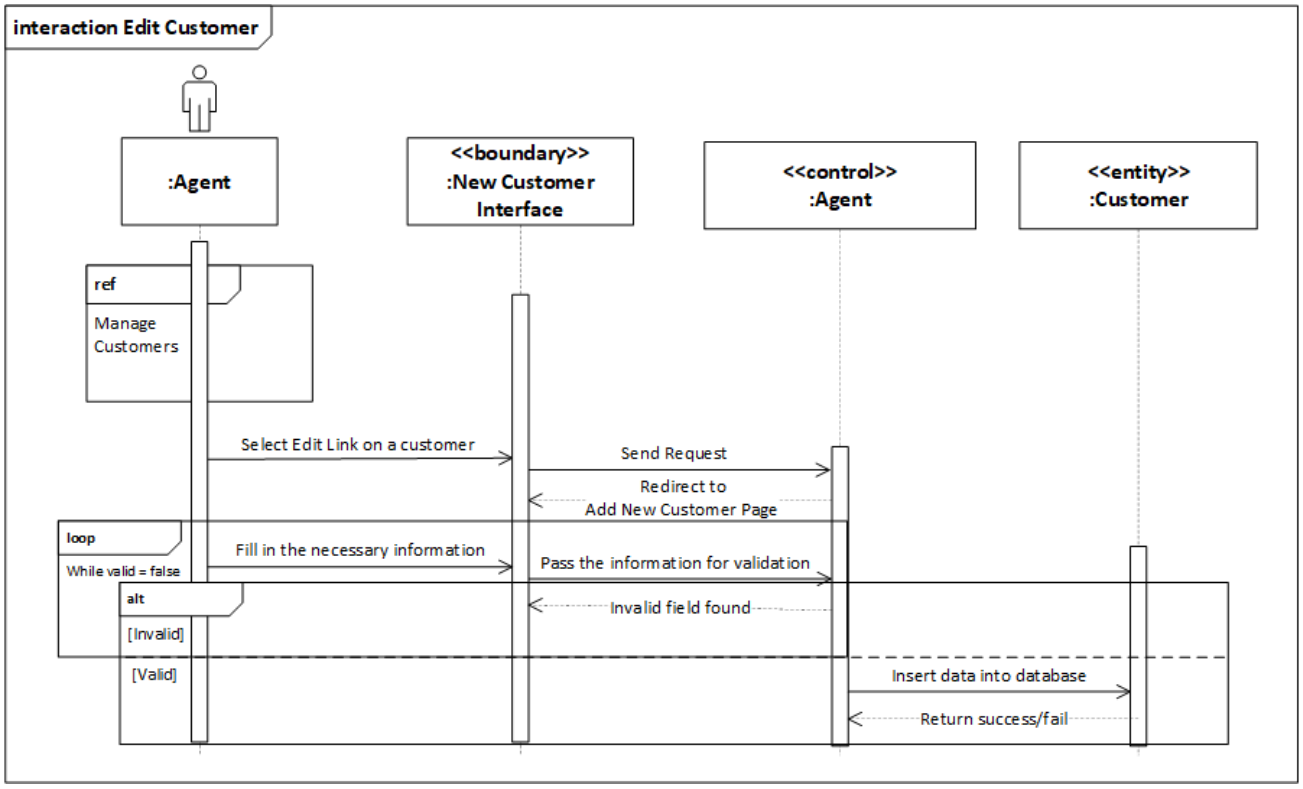
3.2.3.5 Delete Customer



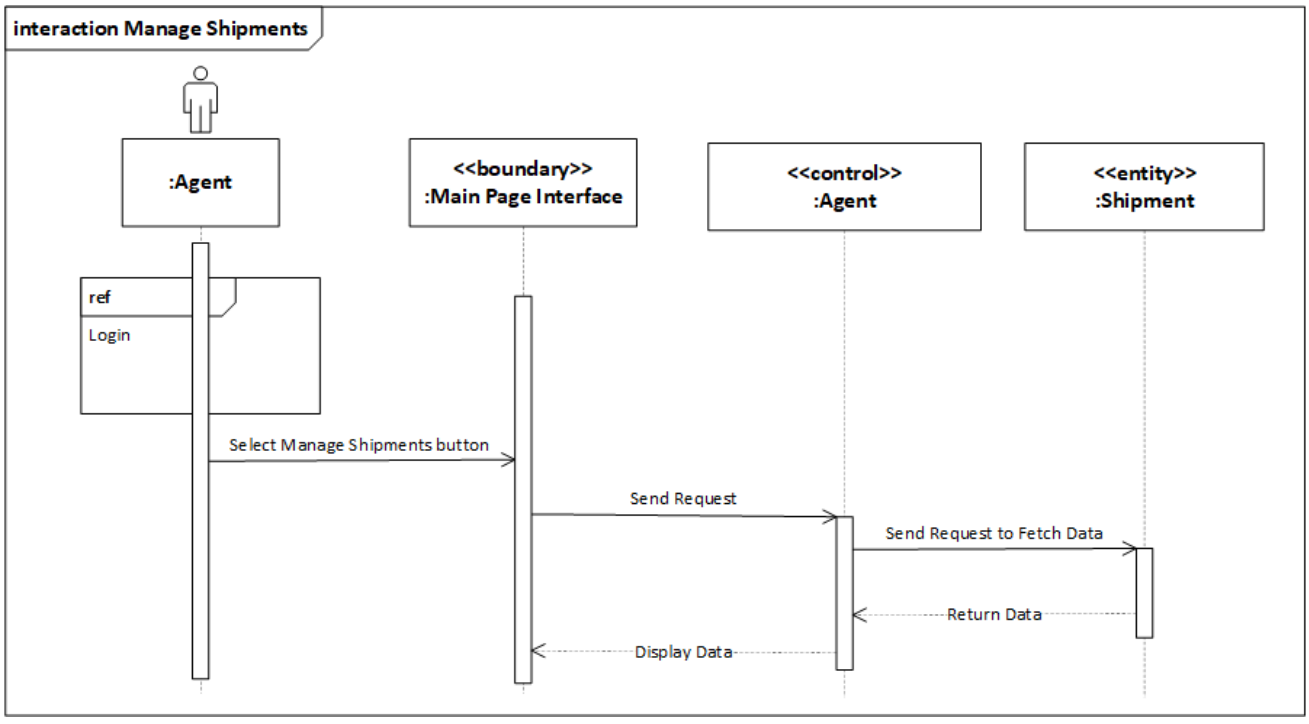
3.2.3.6 Add New Customer



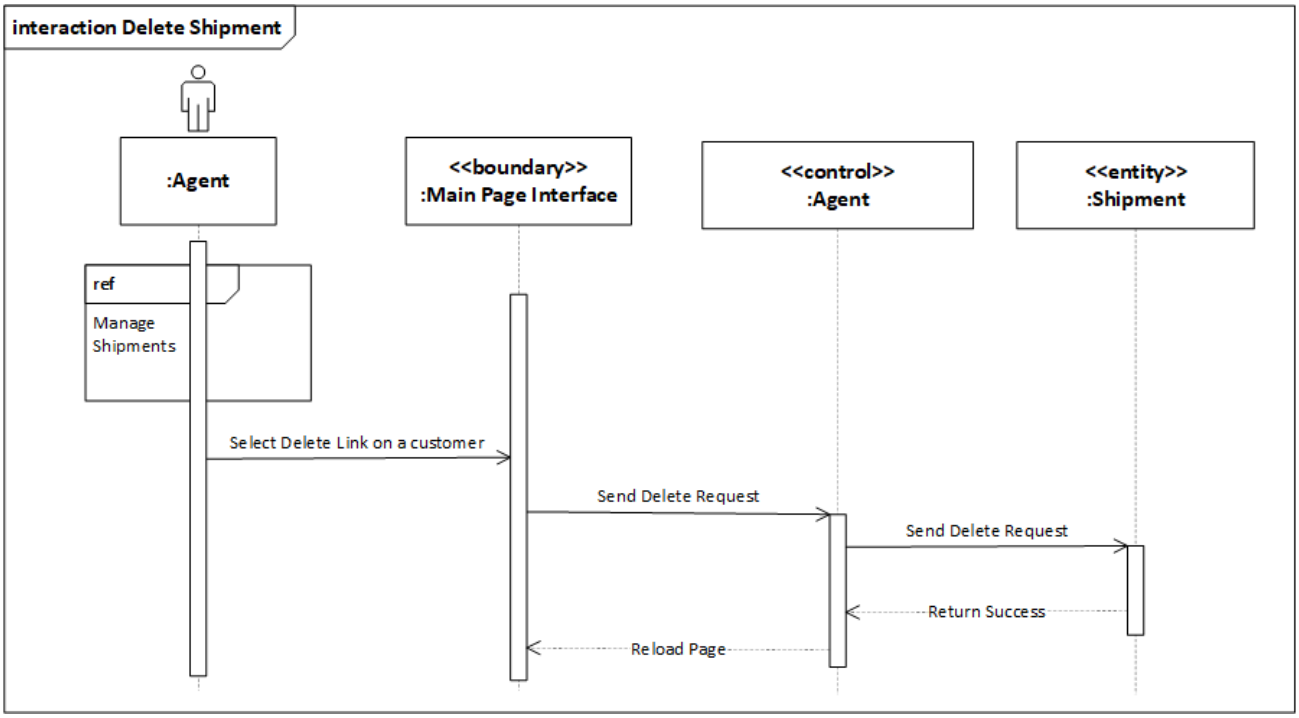
3.2.3.7 Edit Customer



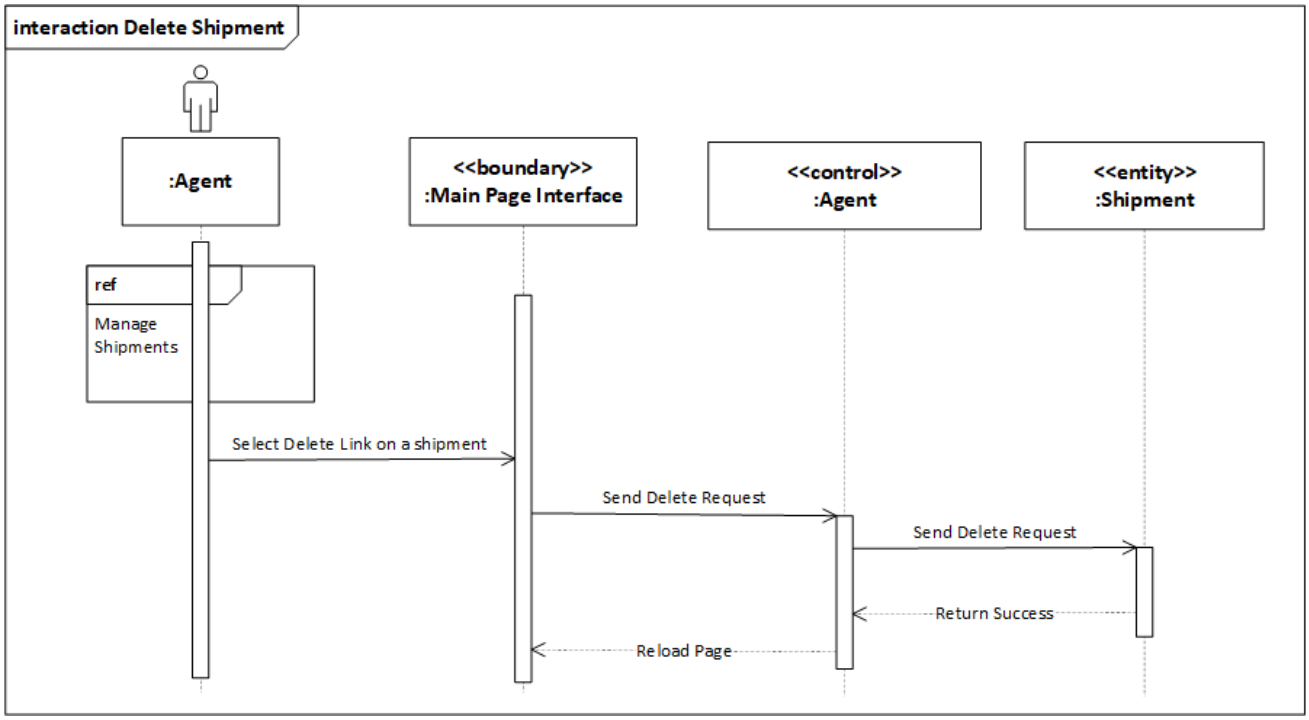
3.2.3.8 Manage Shipments



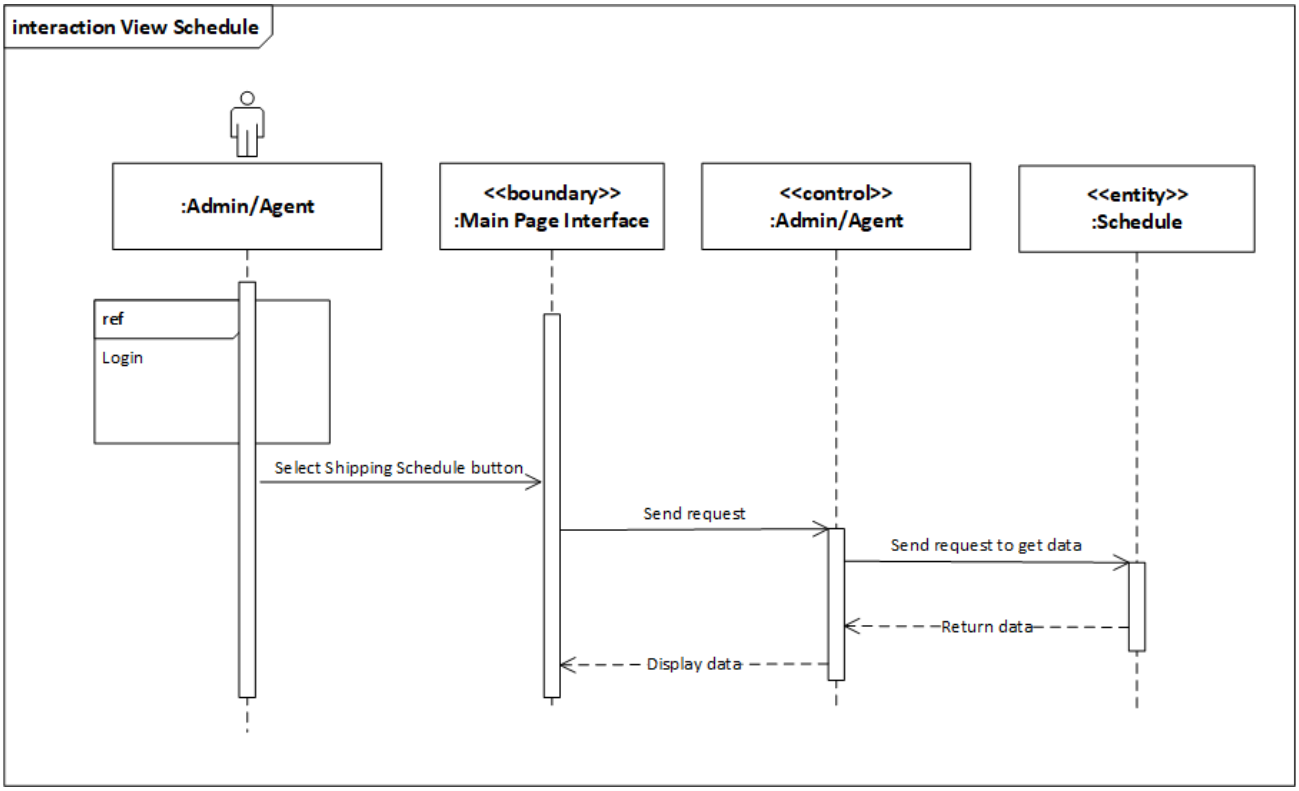
3.2.3.9 Delete Shipment



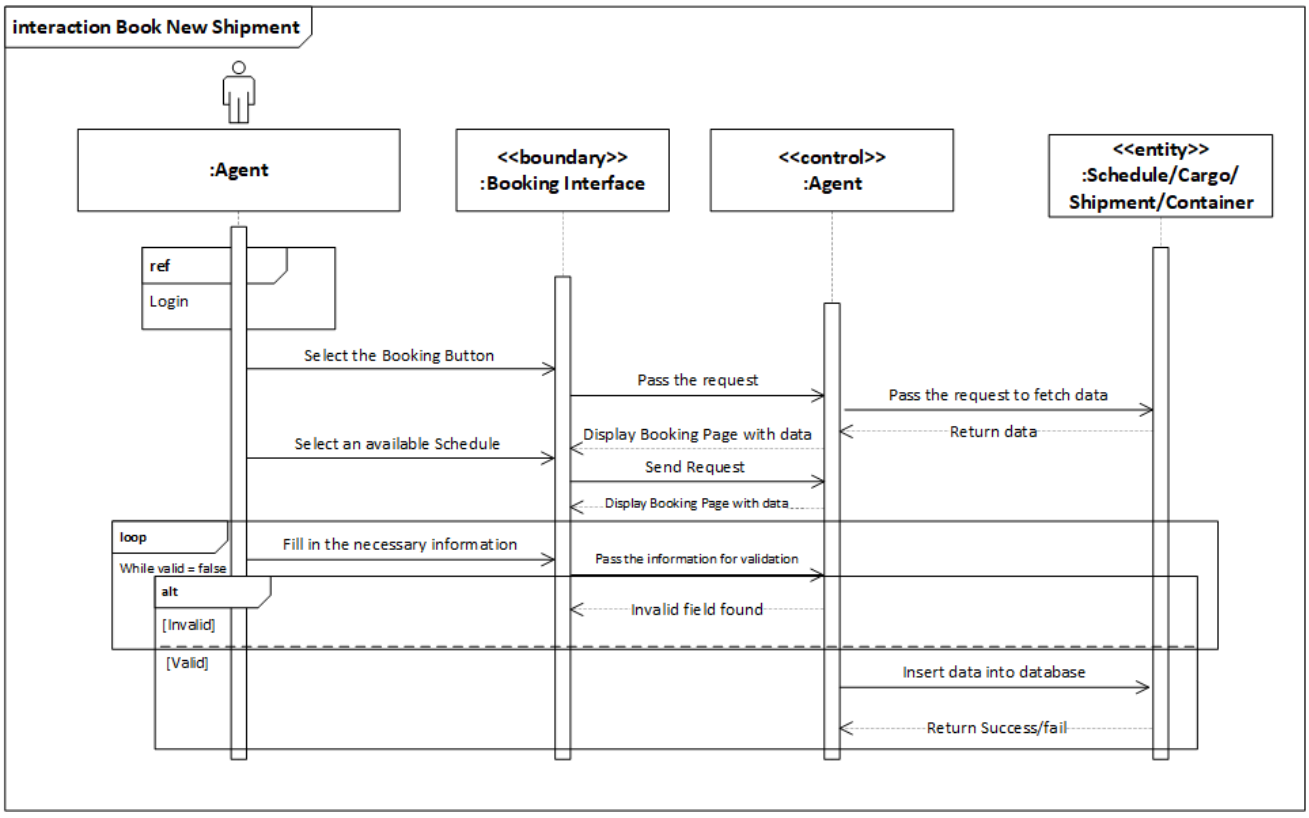
3.2.3.10 View Shipment Details



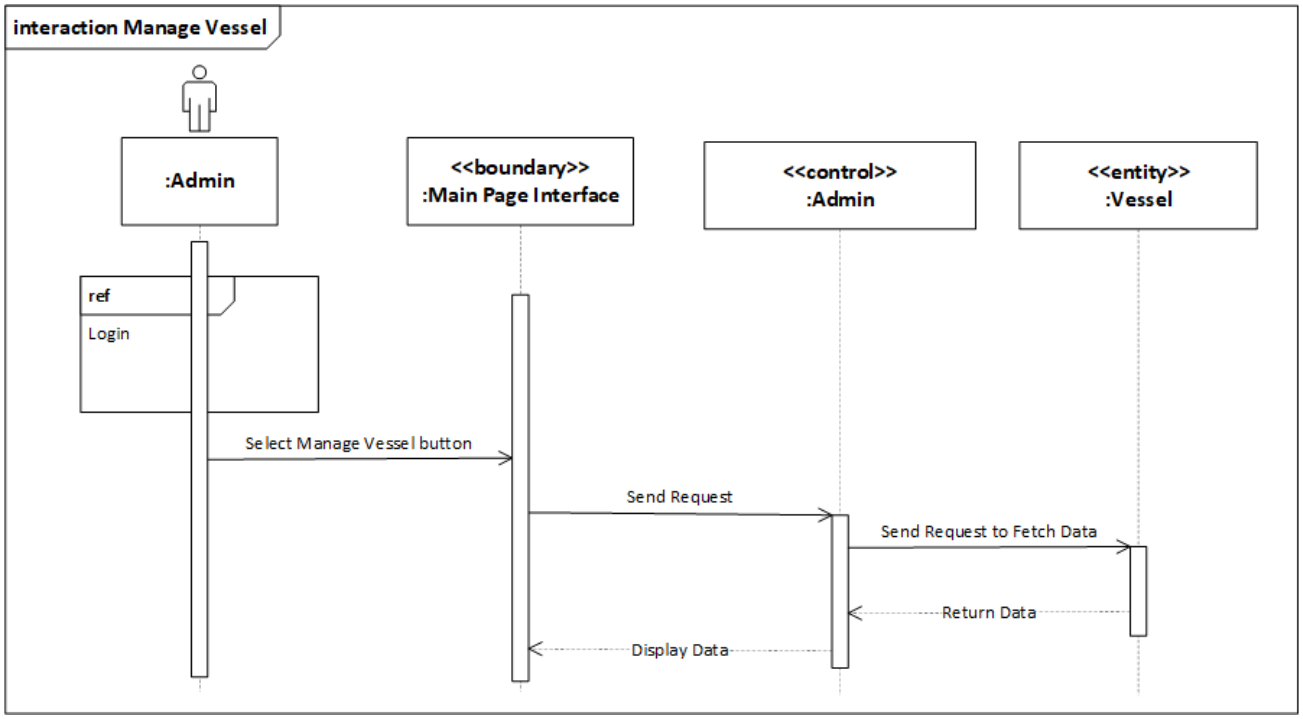
3.2.3.11 View Schedule



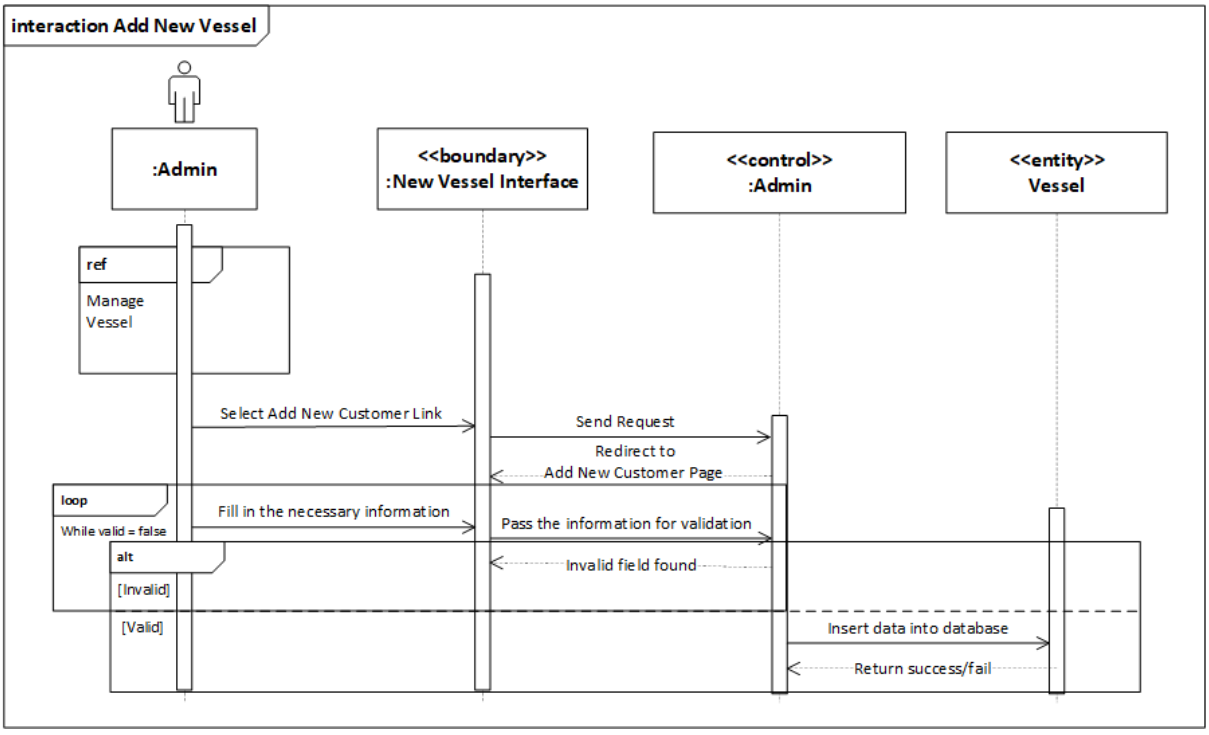
3.2.3.12 Book New Shipment



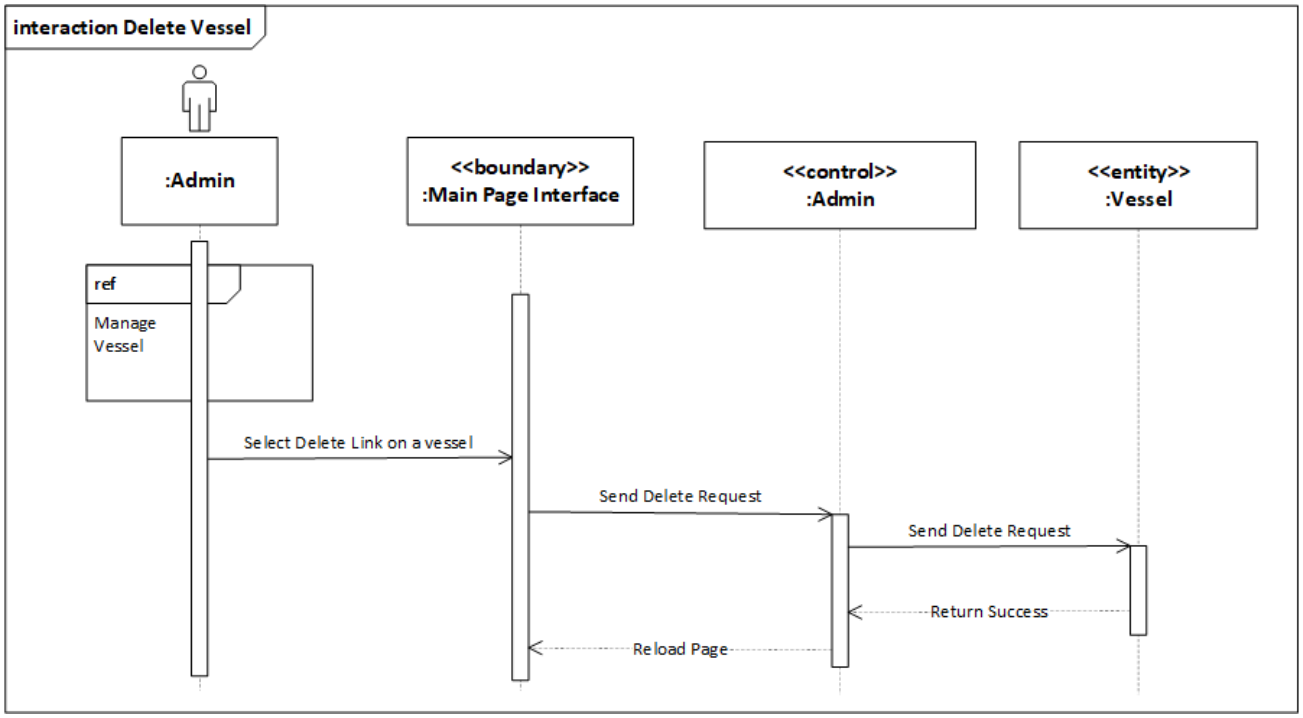
3.2.3.13 Manage Vessel



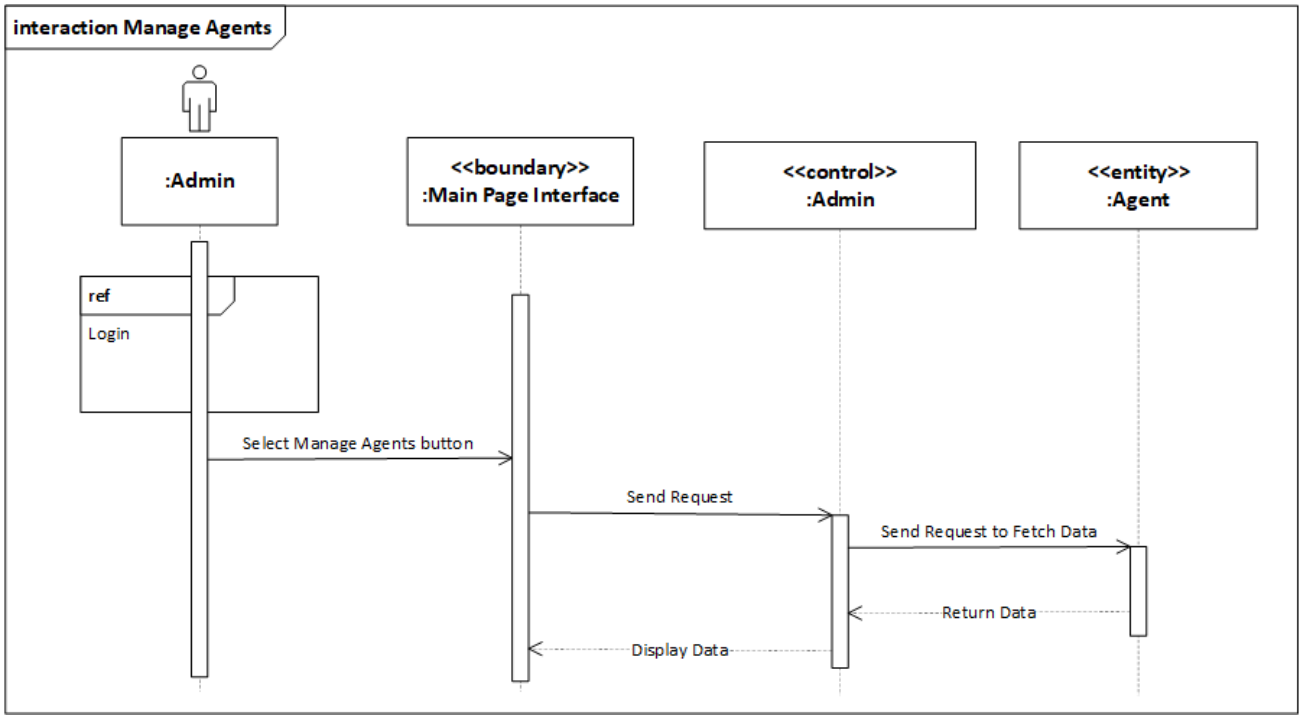
3.2.3.14 Add New Vessel



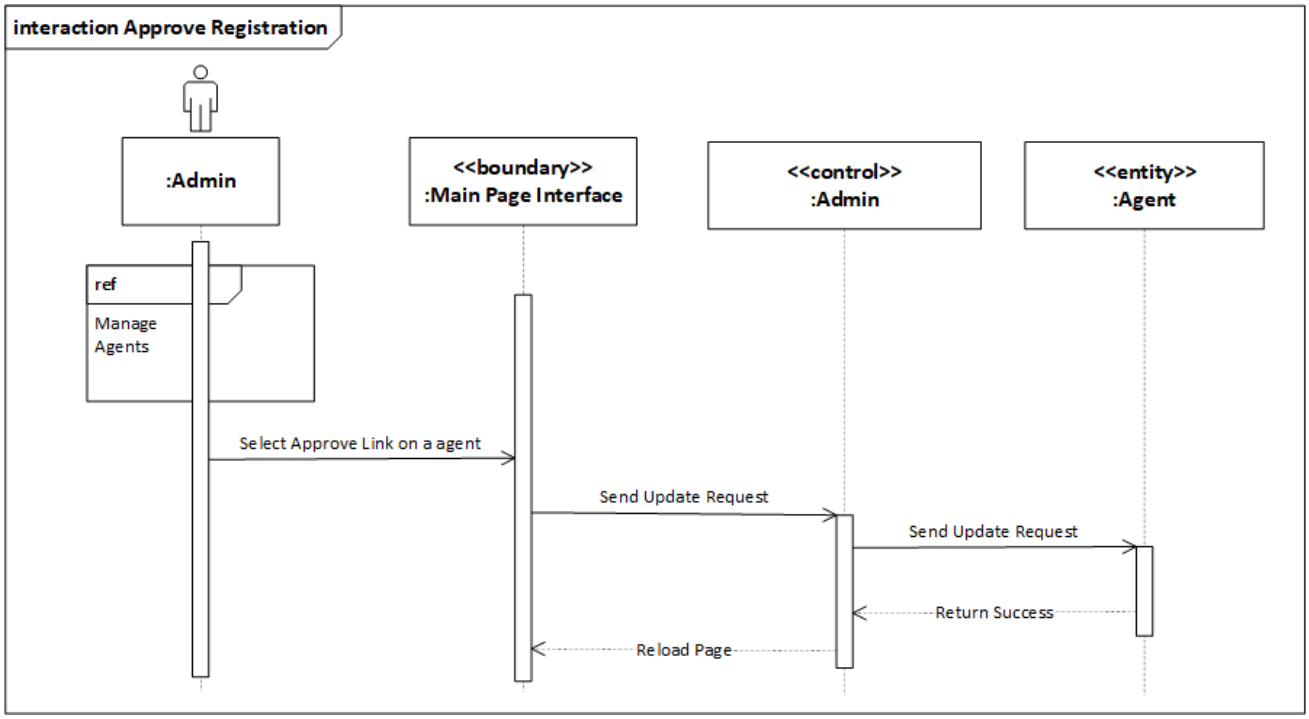
3.2.3.15 Delete Vessel



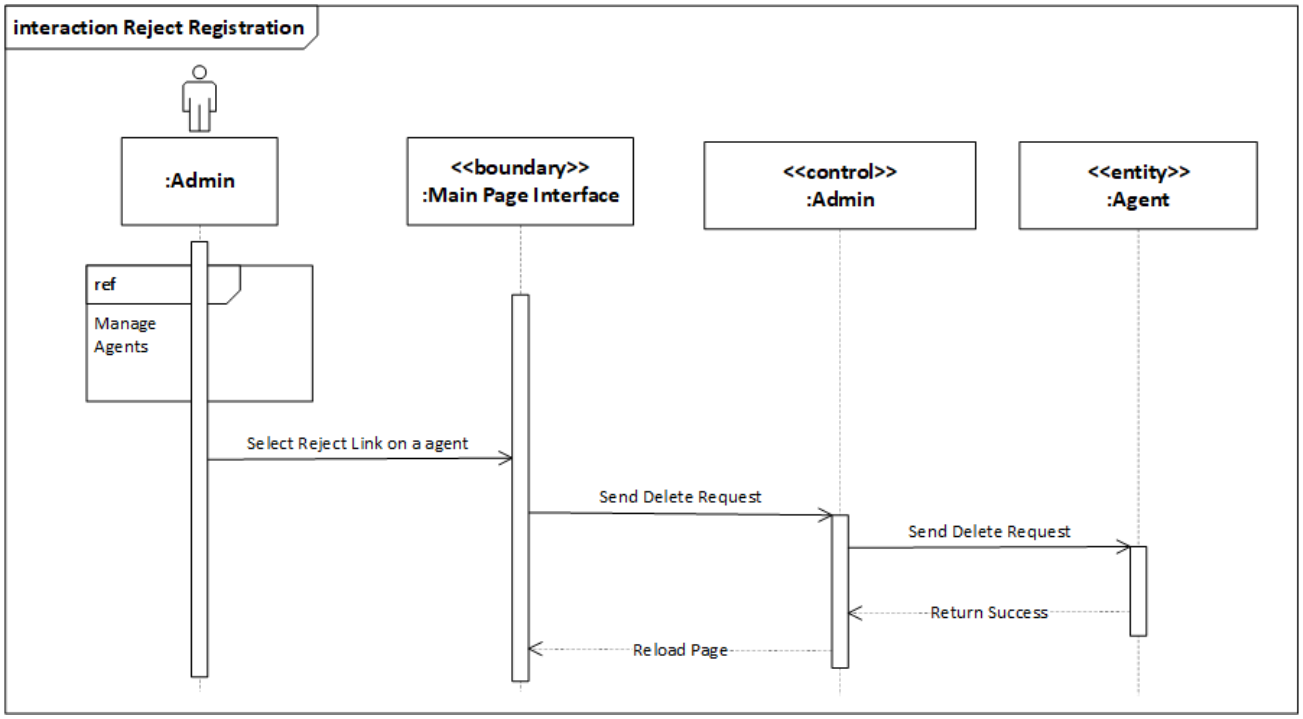
3.2.3.16 Manage Agents



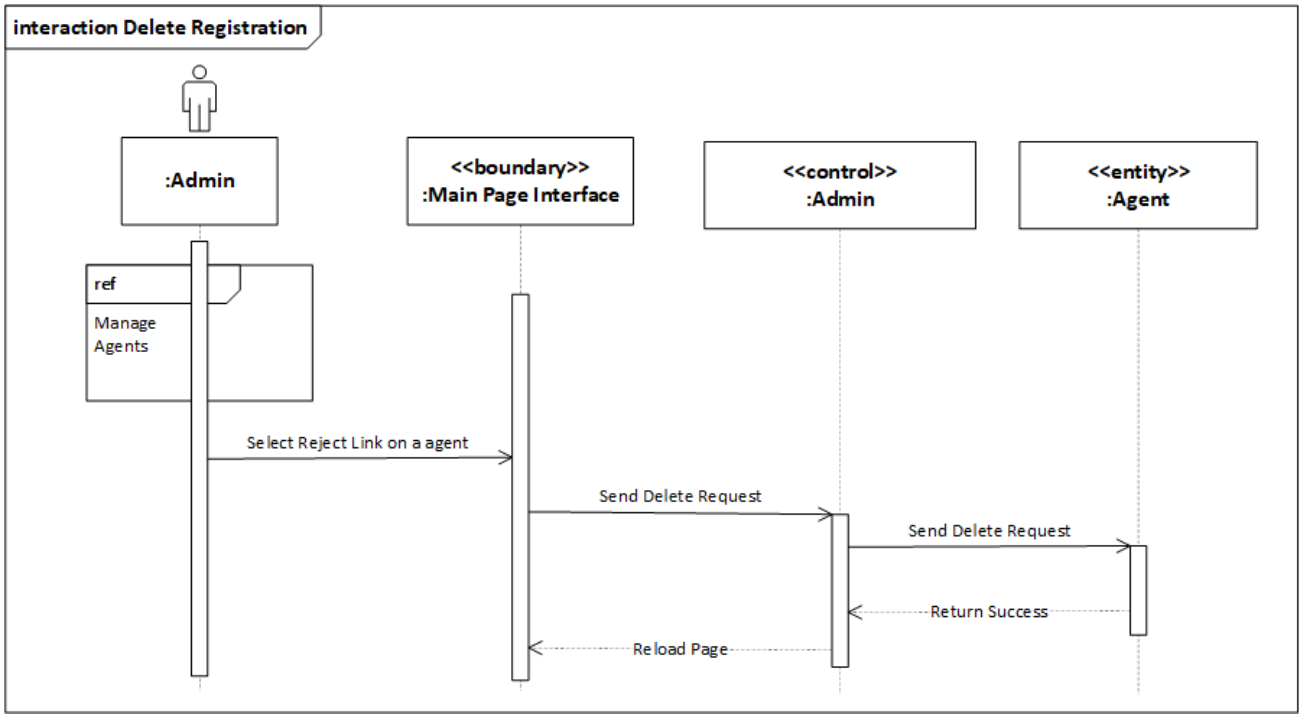
3.2.3.17 Approve Registration



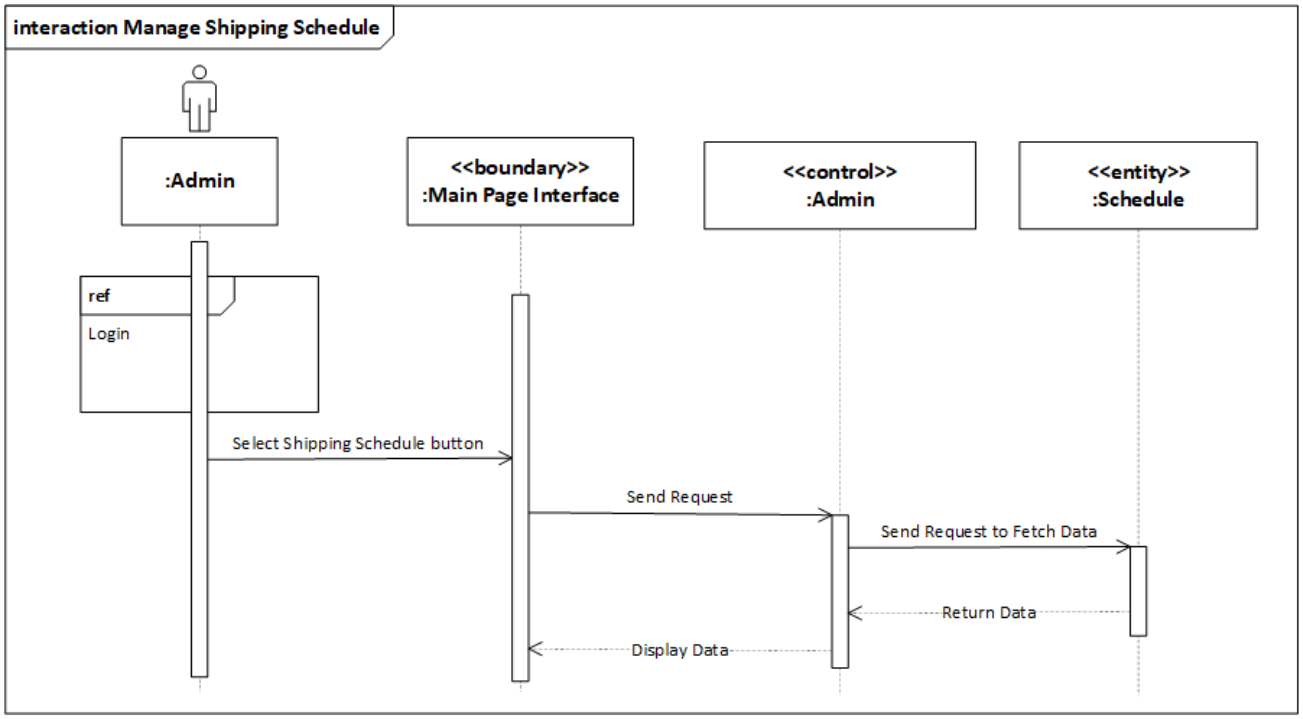
3.2.3.18 Reject Registration



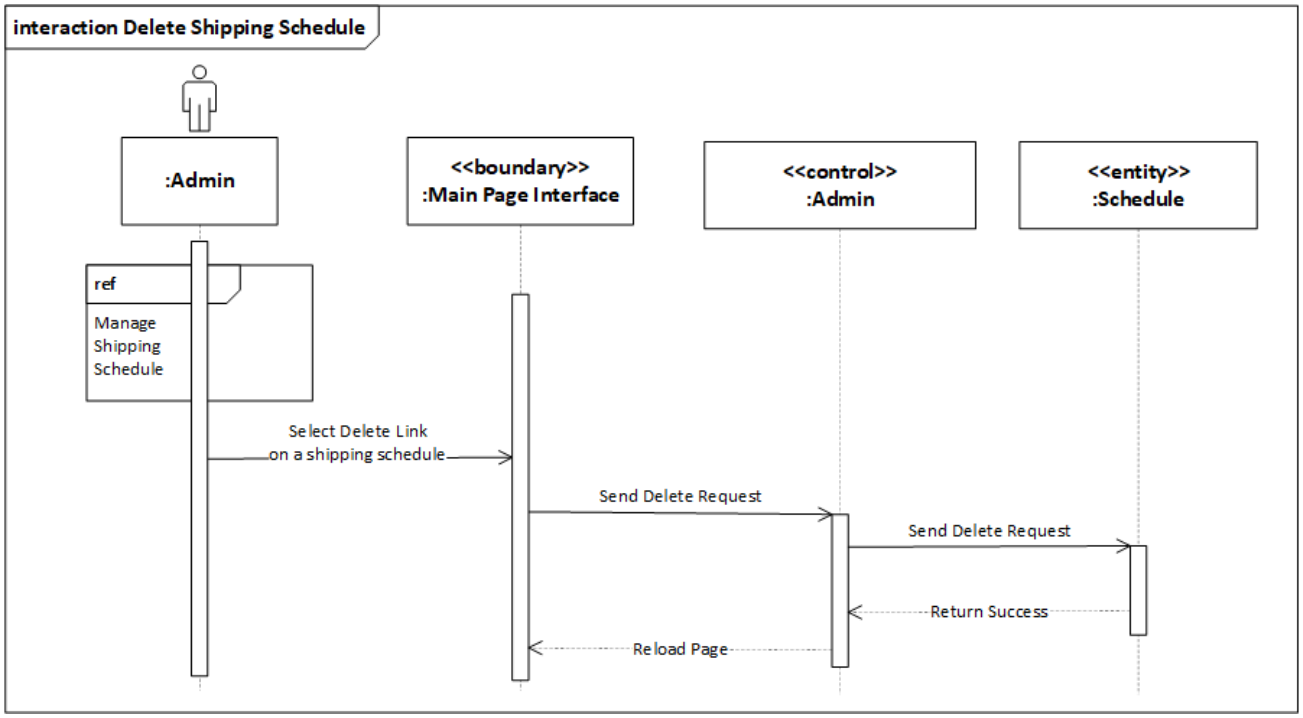
3.2.3.19 Delete Agent



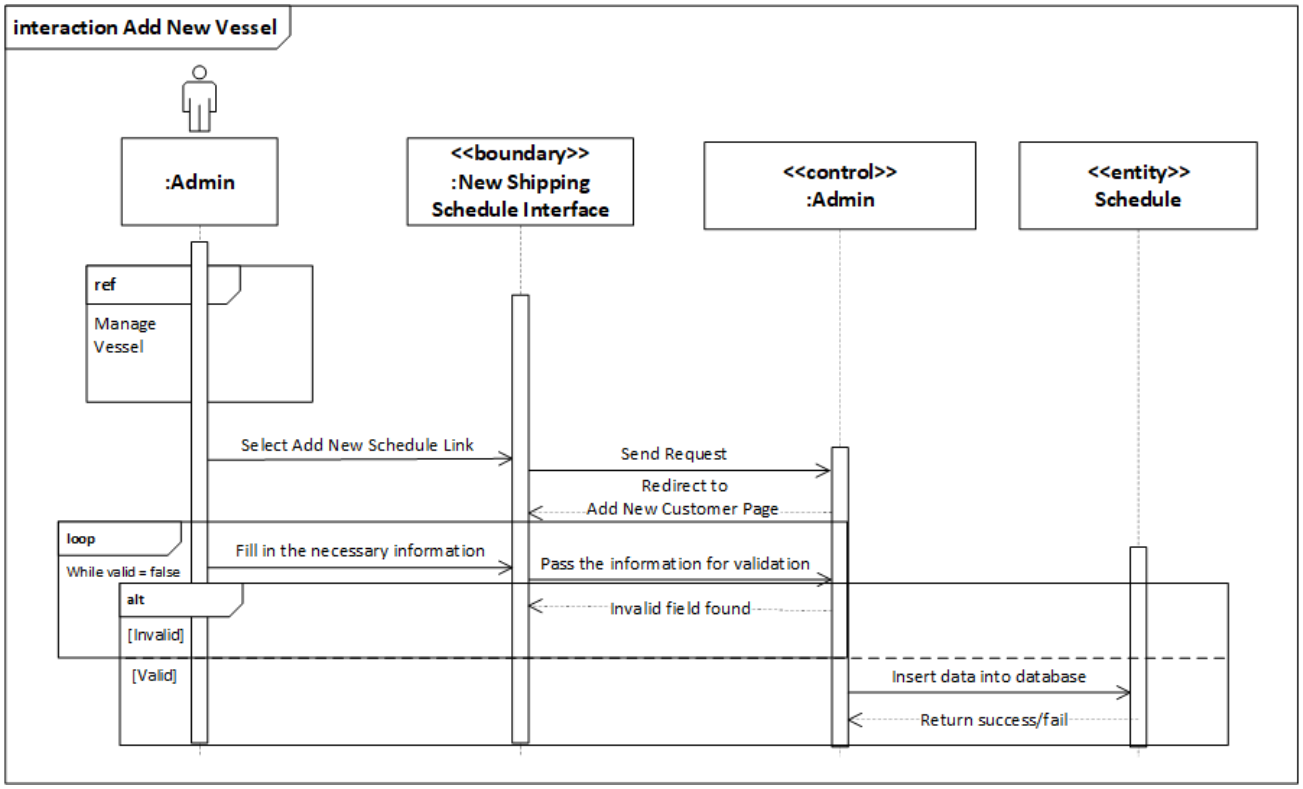
3.2.3.20 Manage Shipping Schedule



3.2.3.21 Delete Shipping Schedule



3.2.3.22 Add New Shipping Schedule



3.3 CLASS DIAGRAM

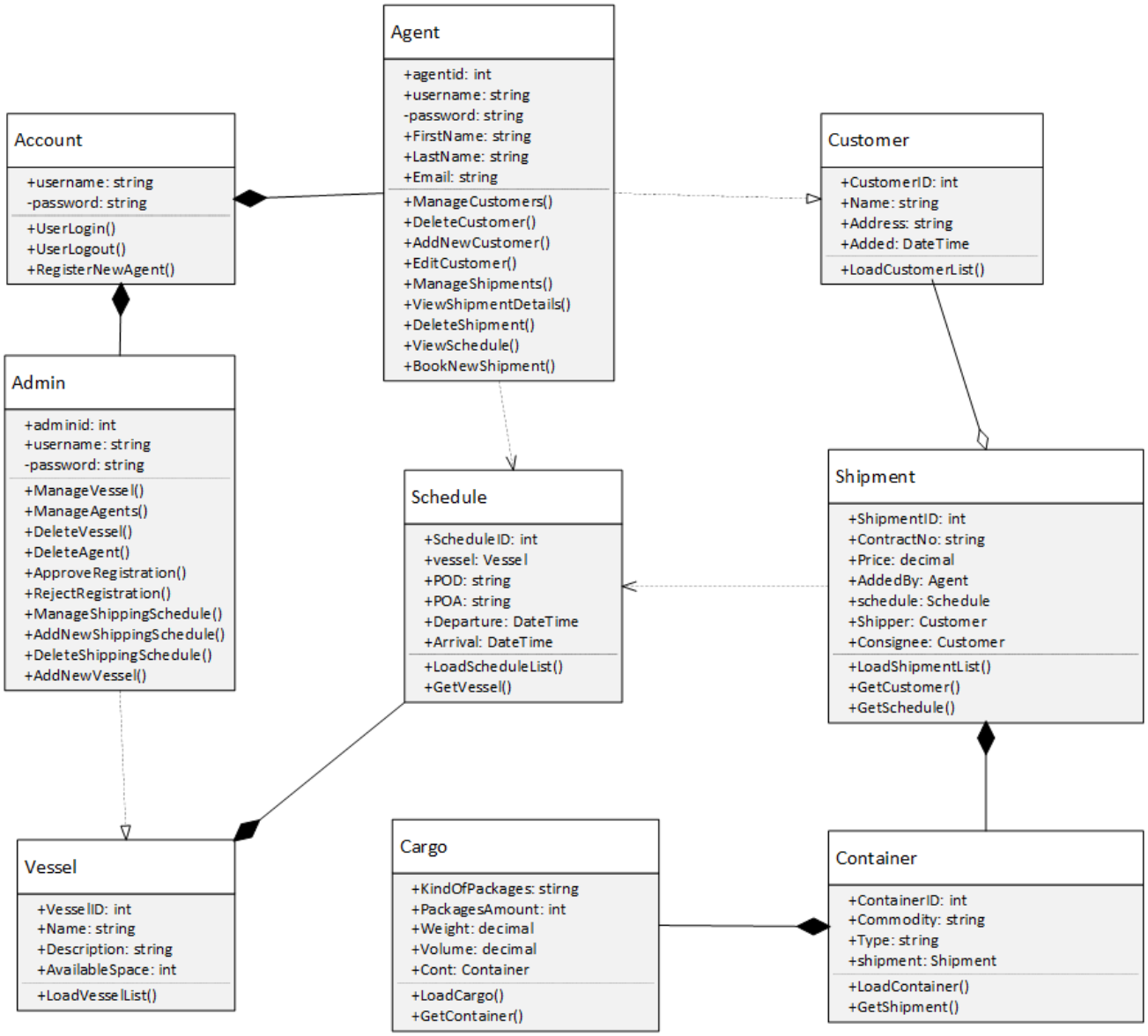


Figure 3-3 - Class Diagram of the Maersk Line CMS

3.3.1 Cloud Architectural Diagrams

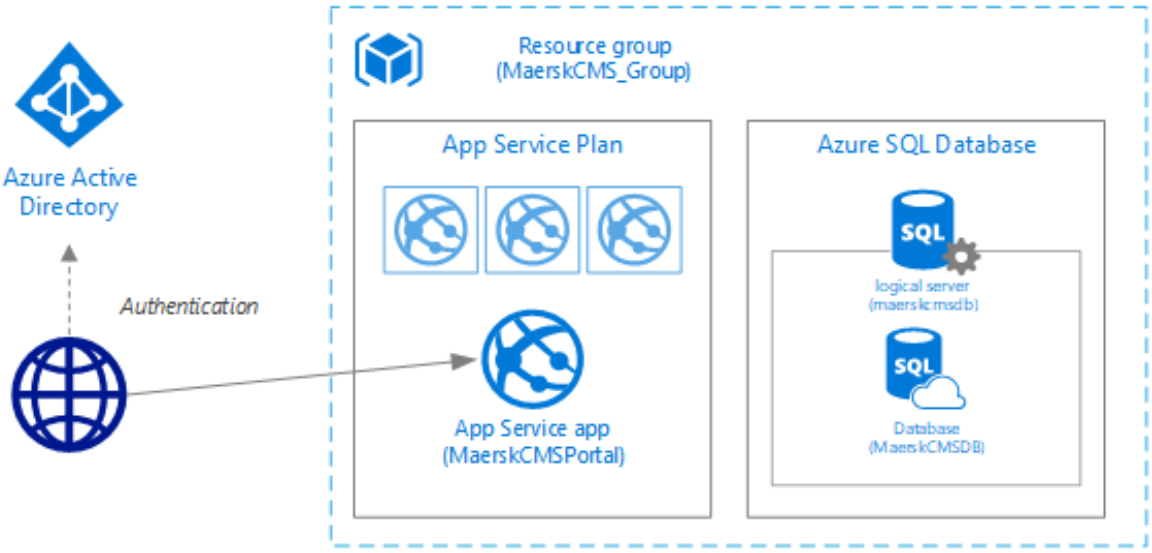


Figure 3-4 - The Cloud Architecture for Development Purpose

The figure above shows the Azure Cloud Architecture diagram when performing the development stage. During the development, the developer configures the Azure Cloud based on the architecture diagram above to allow faster deployment and testing before release.

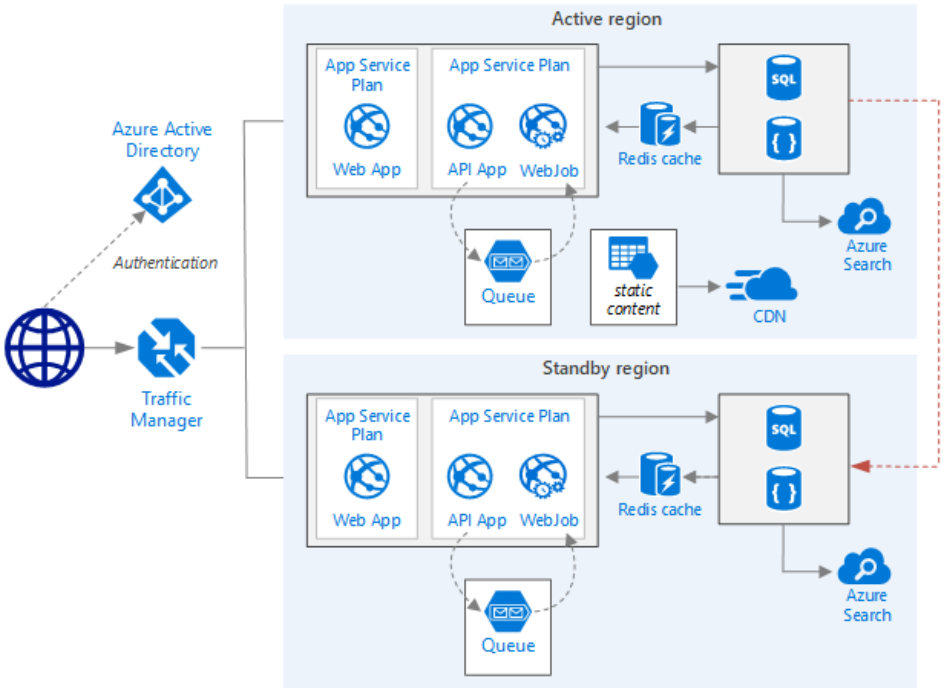


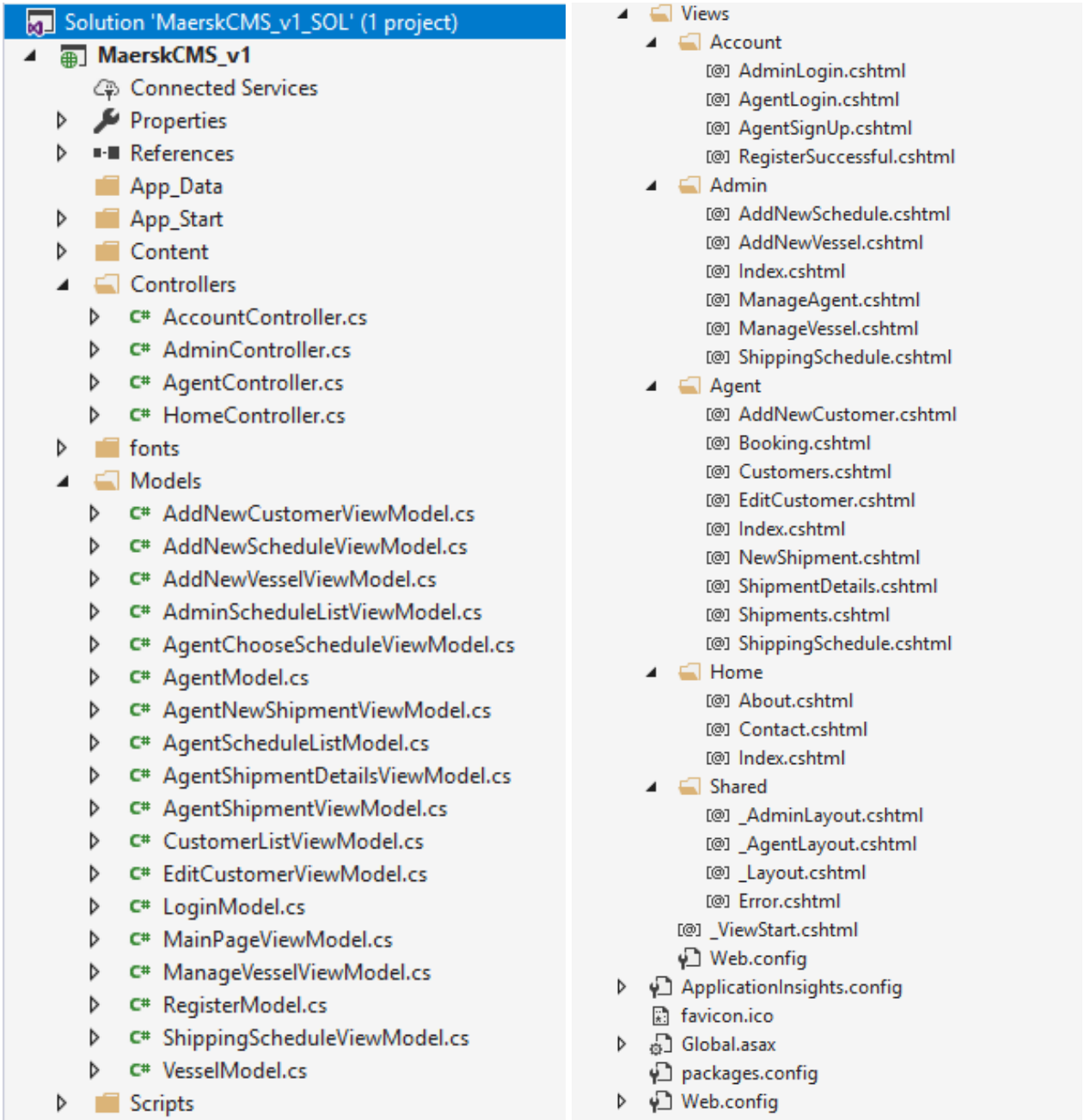
Figure 3-5 - Cloud Architecture after scaling

When the business growth becomes obvious, the figure above shows the Cloud Architecture after scaling up to support real-life operation and simultaneous users across different region.

4. IMPLEMENTATION

4.1 APPLICATION DEVELOPMENT

At the beginning of the implementation of the Maersk Line CMS, the Visual Studio 2017 has been used as the IDE to develop the main application. The Maersk Line CMS is developed with ASP.NET C# with MVC (Model, View, and Controller) structure. The languages that are used to develop the application are C#, Javascript, CSS, and HTML. The structure of the project is presented as shown in the image below.



Besides that, the application is connected to the Microsoft SQL Server to store, update and retrieve data upon requests. The developer uses Microsoft SQL Server Management Studio to create and develop the SQL database according to the developed data model. The developer heavily creates constraints and stored procedures for accessing the data from the SQL Server database.

All the development and implementation are first performed locally at the developer site to ensure the efficient and effective delivery of the proposed system by allowing to perform development, debugging, and testing at ease.

The developer first develops the presentation layer of the Maersk Line CMS for a better visualization on how the pages interlink with each other and also allows the developer to determine the steps of performing the business processes. Next, the developer starts developing the SQL database by creating tables, constraints, relationships and stored procedures and inserts the sample data for testing the data access. Then, the developer starts creating the logics and business processes in the application based on the initial requirements. The testing and debugging are performed during this phase.

Upon launching the first page of the Maersk Line CMS, the homepage of the Maersk Line CMS is displayed. The page contains the name of the system, a navigation bar, and some options to starts with.

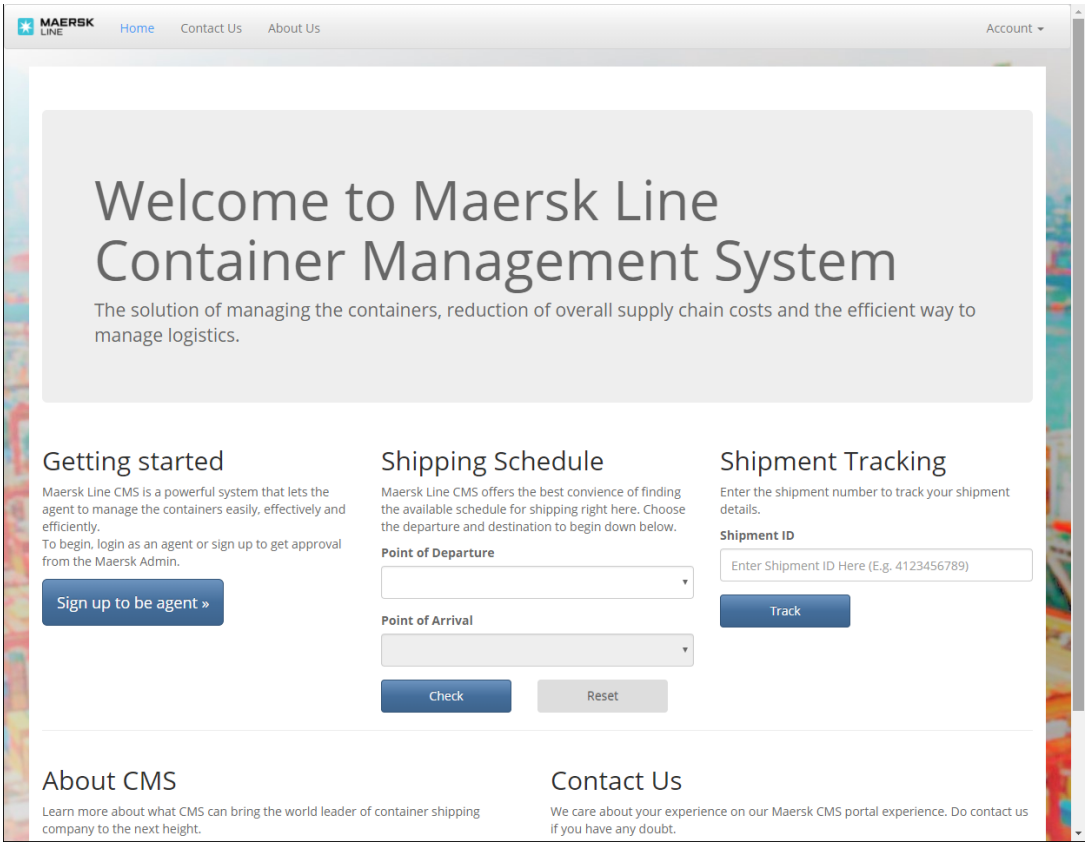


Figure 4-1 - User Interface for Guest Home Page

The user access control by validating the credential is done to prevent unauthorized access to the unappropriated area of the system. To register as an Agent, the user must first register through the Register page and wait for approval from the administrator before using the services in the Maersk Line CMS. The role of the Admin is to manage vessels, manage schedules and also manage the agents with approval process. However, the role of the Agent is to manage customers, perform a booking of a shipment, check the schedule and manage the existing shipments associated with the respective agents.

The source code of the system will be available at this link: https://github.com/kelvinyap5673/DDAC_TP037636/ A full demo of the system functionality will be included as a video in this link: https://web.microsoftstream.com/video/aaeac2ae-8ff3-406e-ad6d-0b83c89c48ae.

4.2 AZURE PUBLISHING

4.2.1 Create New Resource Group on Azure Portal

In this first step, the developer creates a new resource group as a container of the Web Application and the SQL Server of the proposed system before the deployment. The resource group is named as MaerskCMS_Group and the resource group location is set to Southeast Asia.

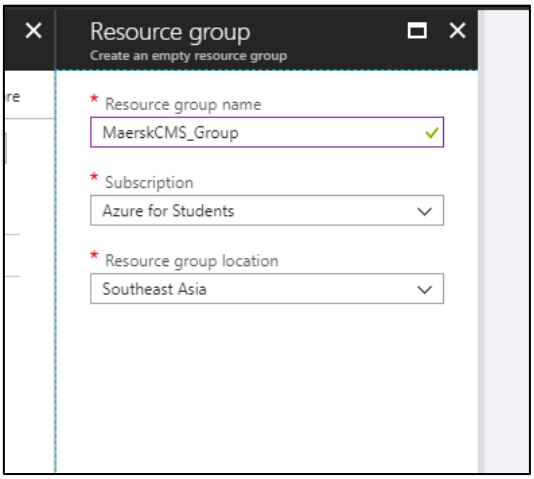


Figure 4-2 - Creating a new empty resource group

4.2.2 Add a resource – Web App + SQL

As the system is developed in ASP.NET C# and SQL, the developer has chosen the Web App + SQL resource for the ease of deployment.

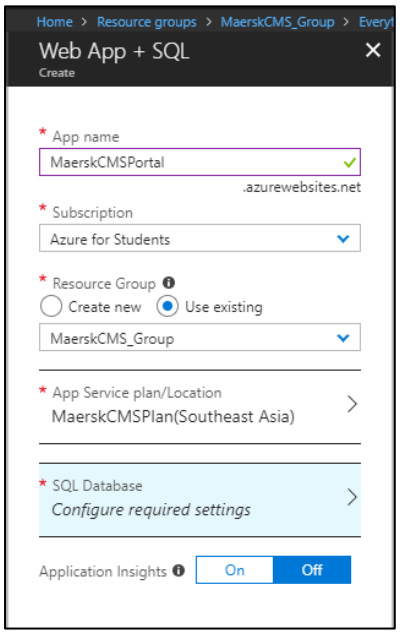


Figure 4-3 - Creating a Web App + SQL

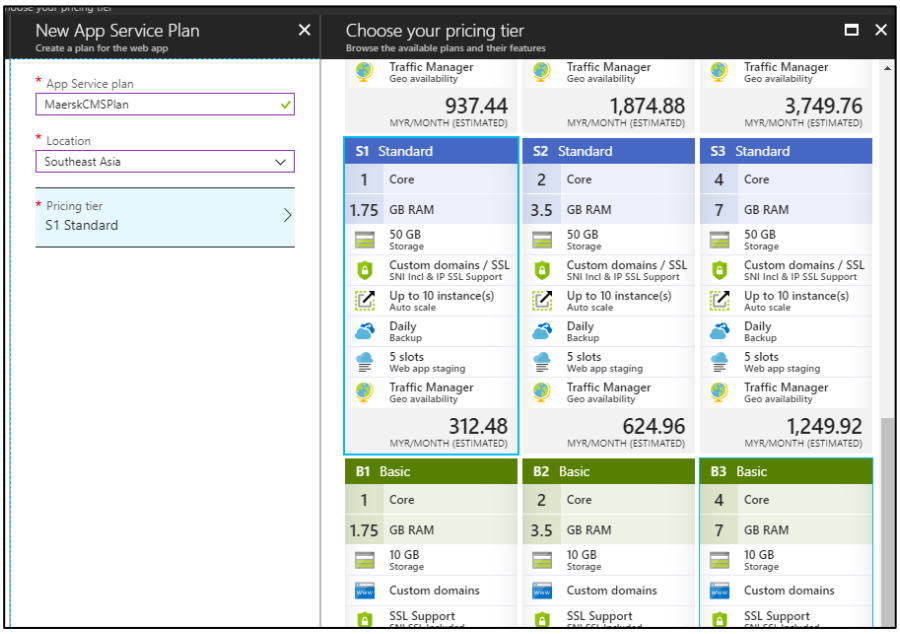


Figure 4-4 - Creating new App Service Plan

The App name is MaerskCMSPortal and uses the existing resource group which is MaerskCMS_Group. Then, the developer has created an App Service plan with the chosen pricing tier of S1 Standard or higher to meet the minimum requirement for having features such as traffic manager , reliability and scalability to support the web application, which will be discussed in the following section 4.3.

In the meantime, the SQL Database is also setup during this stage by adding a new SQL server with details entered, configuration and chosen pricing below.

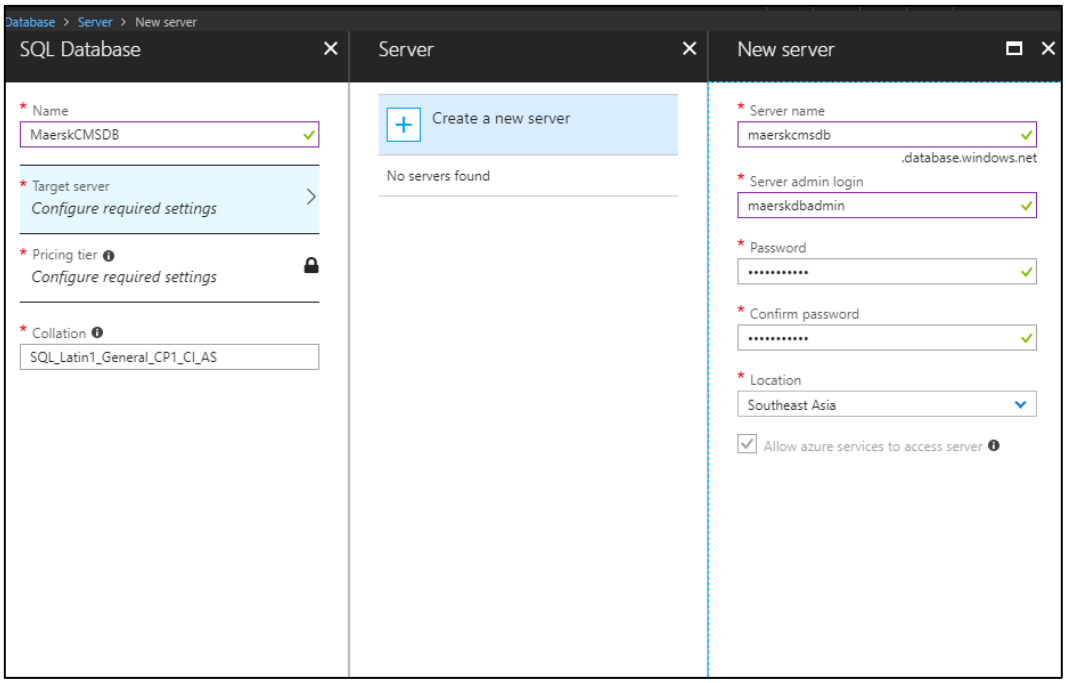


Figure 4-5 - Creating new SQL Database

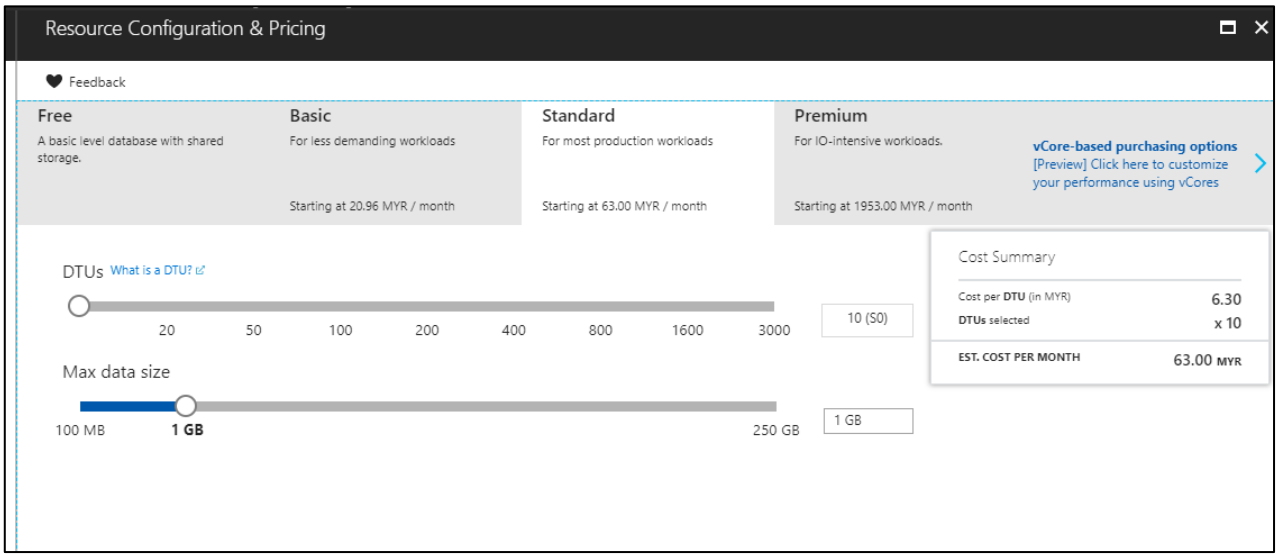


Figure 4-6 - SQL Database Configuration and Pricing choice

At the end, the developer creates the resource with the configured details below.

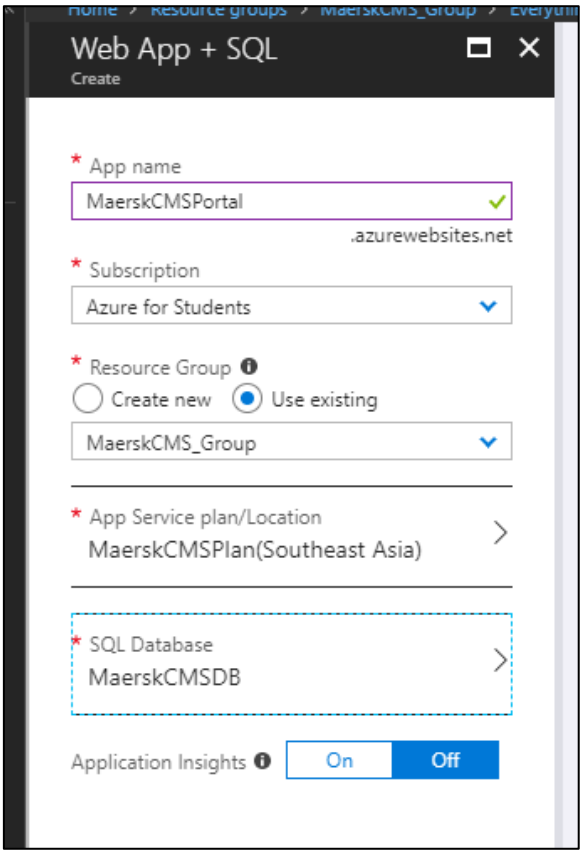


Figure 4-7 - The configured details of the Web App + SQL

4.2.3 Migration of the local SQL Database to the Azure SQL Database

In this step, the Data Migration Assistant, a tool provided by Microsoft, is used to allow data migration from the local SQL database to the created Azure SQL Database in the previous step. First, the developer created a new project to start the migration configuration process.

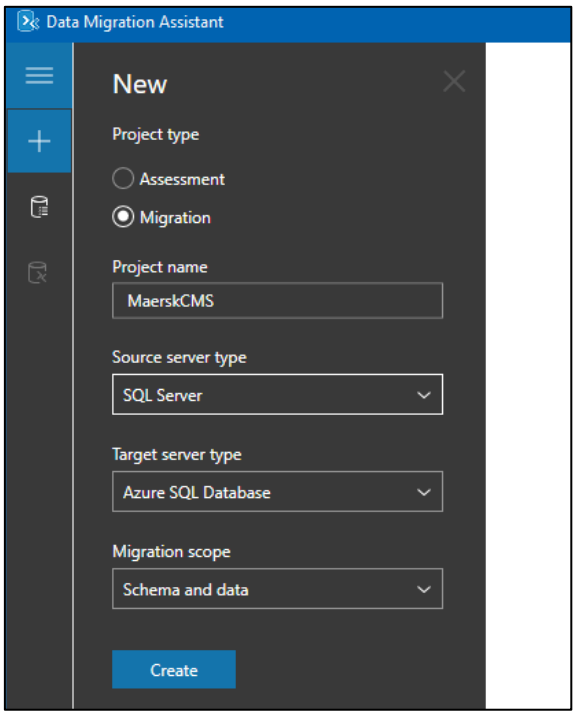


Figure 4-8 - Create a New Data Migration Project

Next, the developer connects the local SQL database and select the desired database to be migrated.

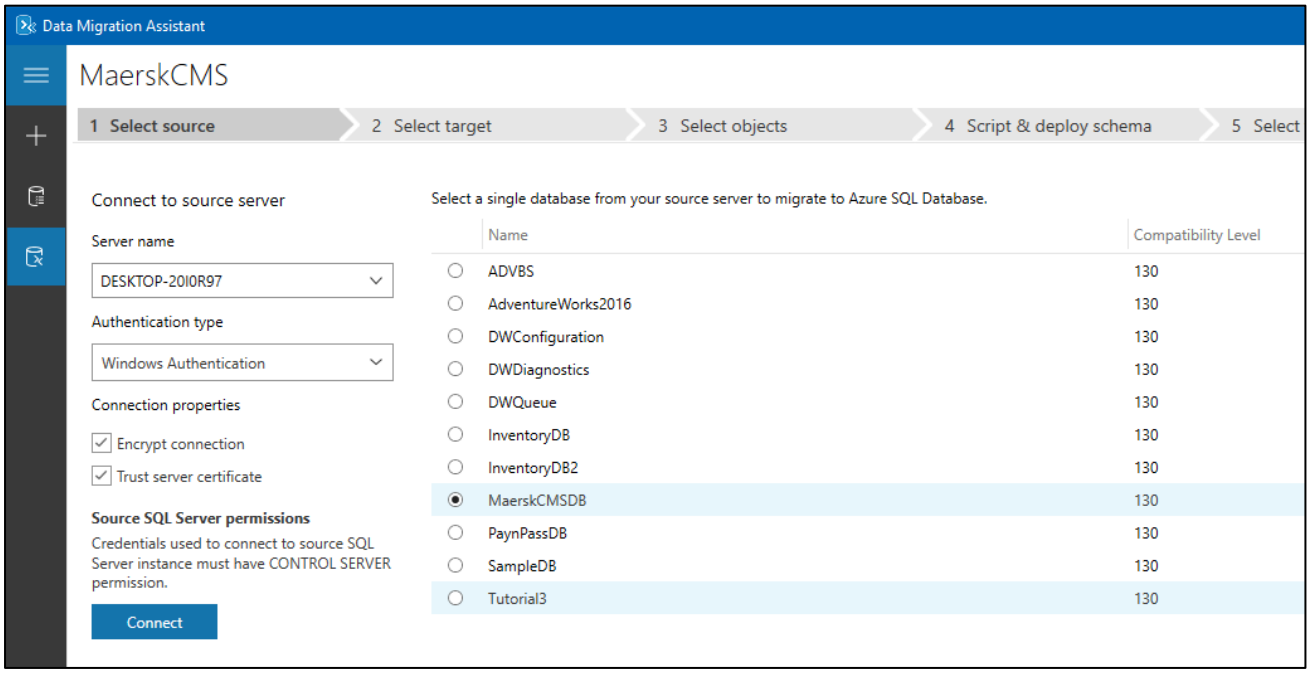


Figure 4-9 - Select source of the database

Then, the developer connects to the target database server which is at the created Azure SQL Database Server.

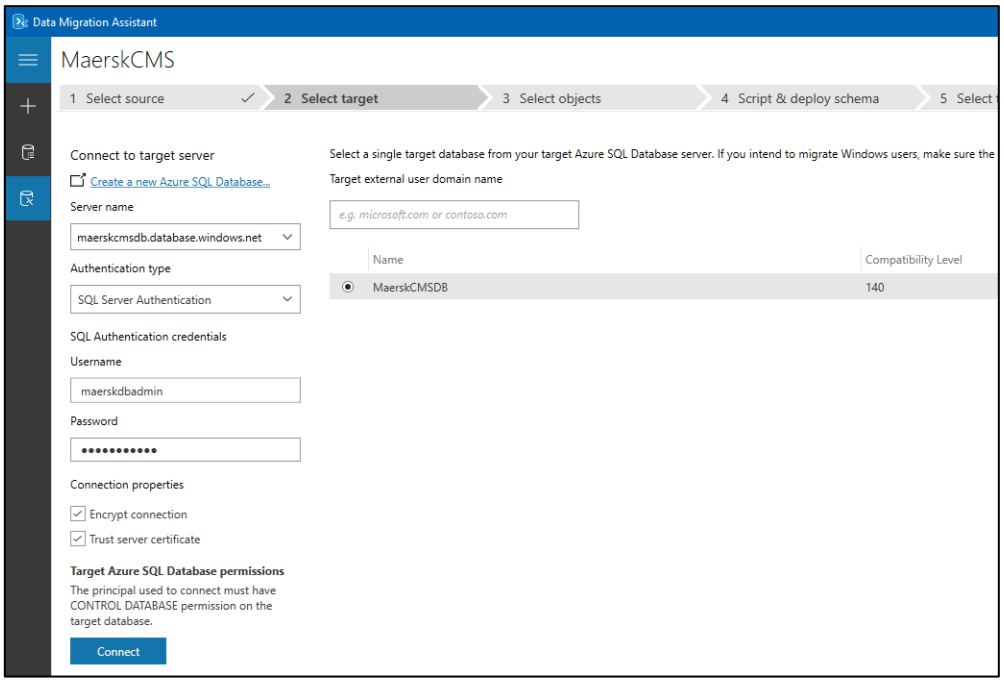


Figure 4-10 - Select the target to migrate to.

After that, the developer select the required objects to be migrated to the target database.

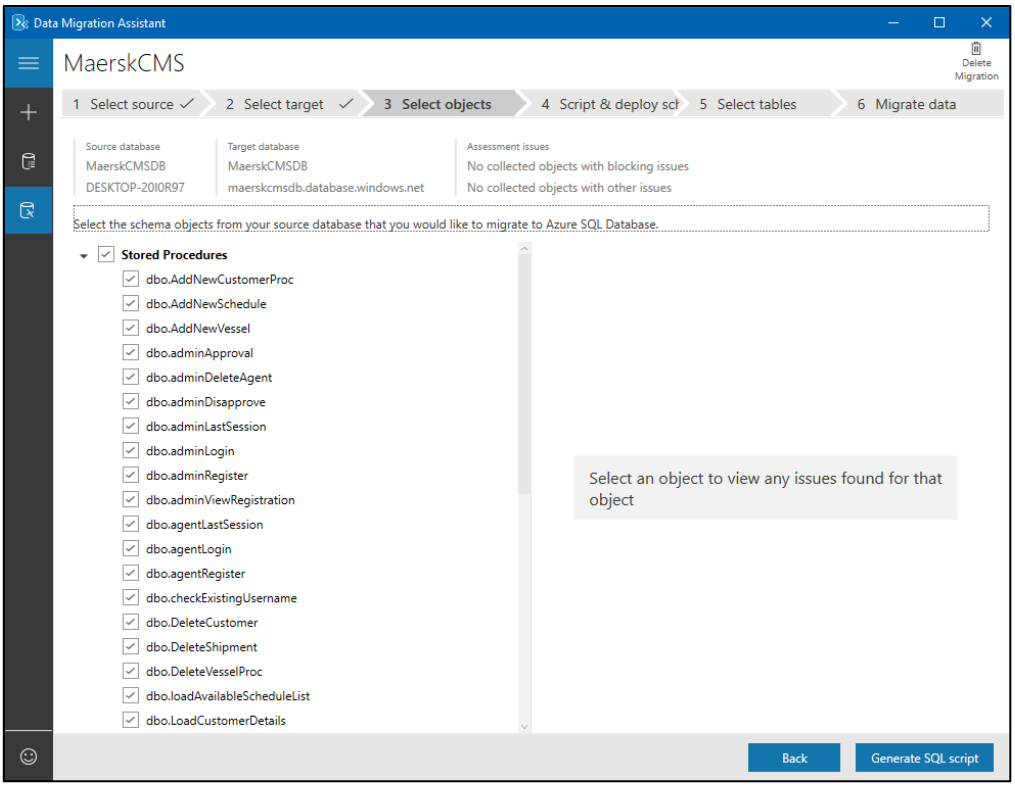


Figure 4-11 - Selecting the objects to be migrated

After selecting the required objects, the Data Migration Assistant will generate a script that allows the target SQL database server to perform the query and also deploying the schema to the target.

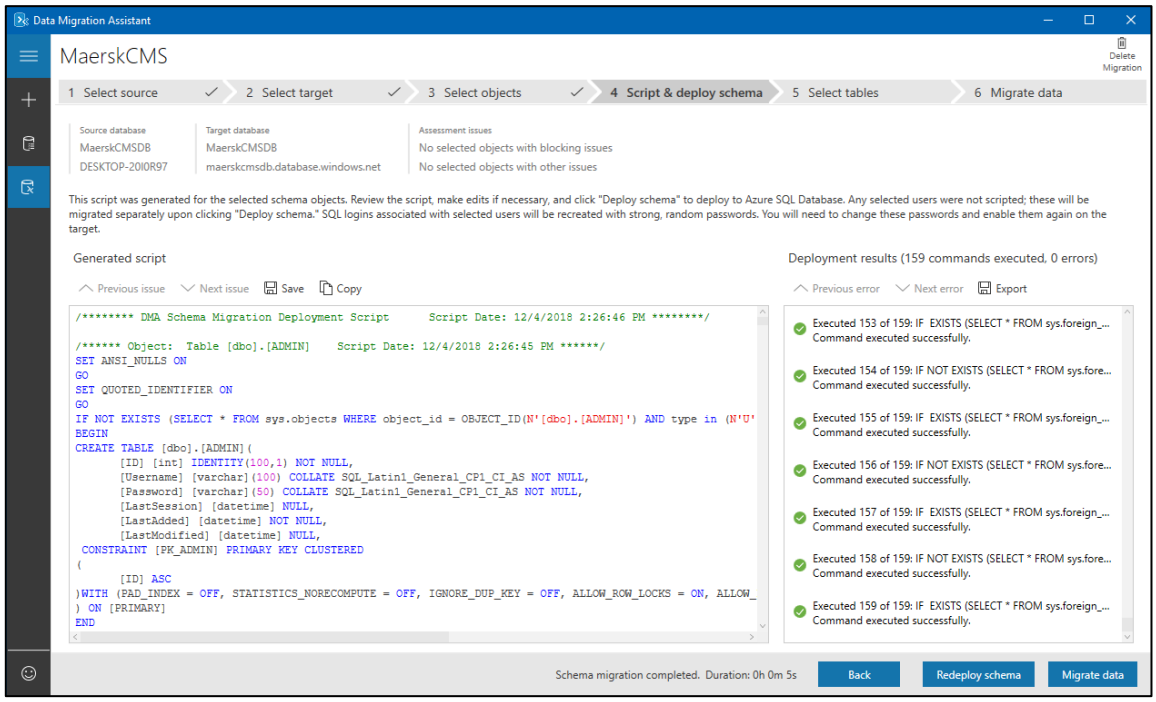


Figure 4-12 - Generated Scripts and Schema Deployment

In the next process, the developer has chosen the required tables to be deployed to the target.

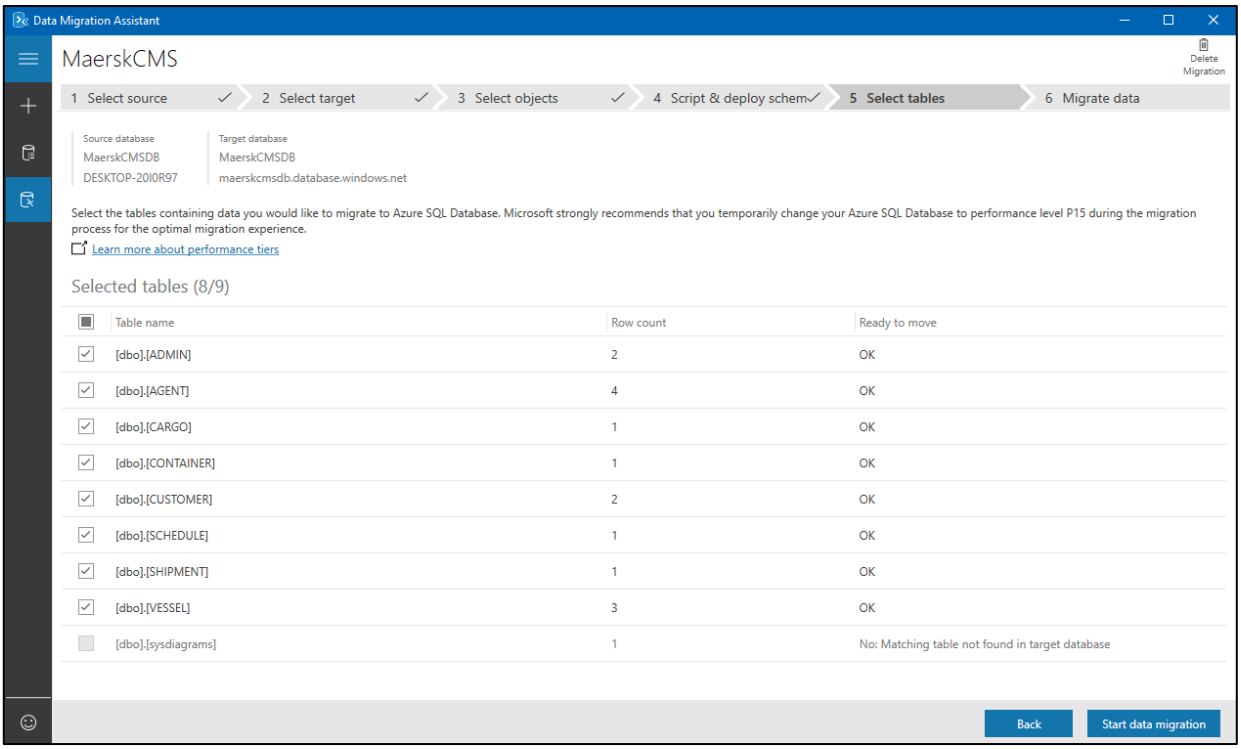


Figure 4-13 - Selecting the required table.

At last, the developer initiates the data migration process. After the data migration completed, the result of the migration is shown below.

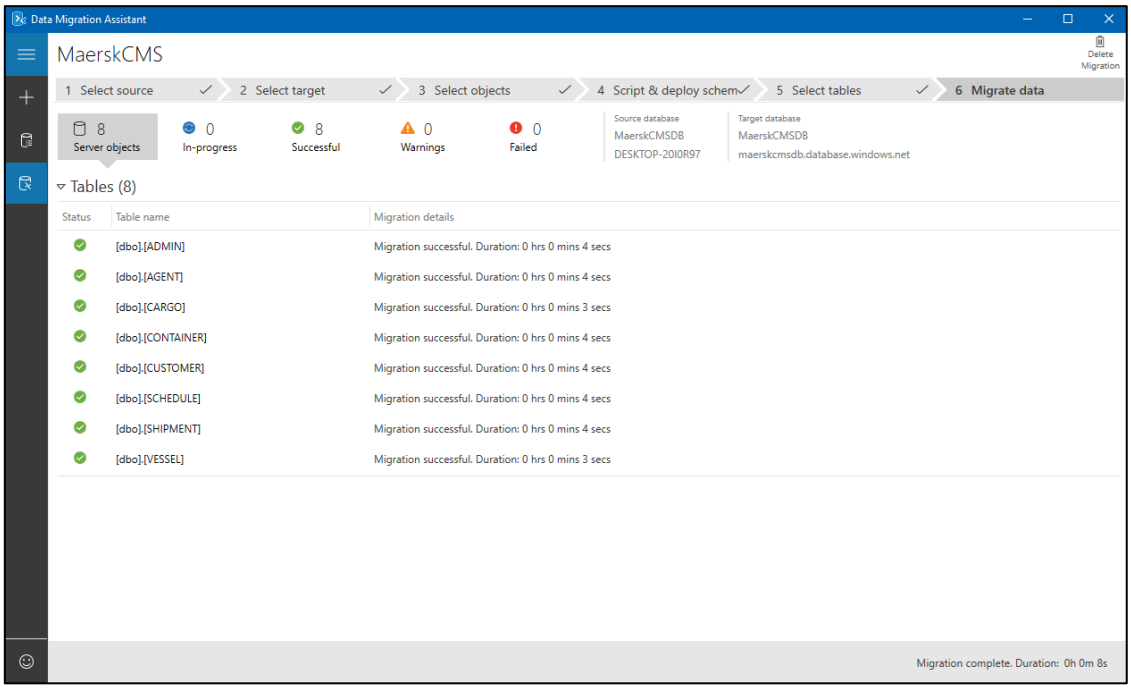


Figure 4-14 - The result of the data migration after attempt.

4.2.4 Publish the Web Application

In this step, the developer starts to deploy the Maersk Line CMS to the Azure. At first, the developer right-clicked on the project and click on the Publish option.

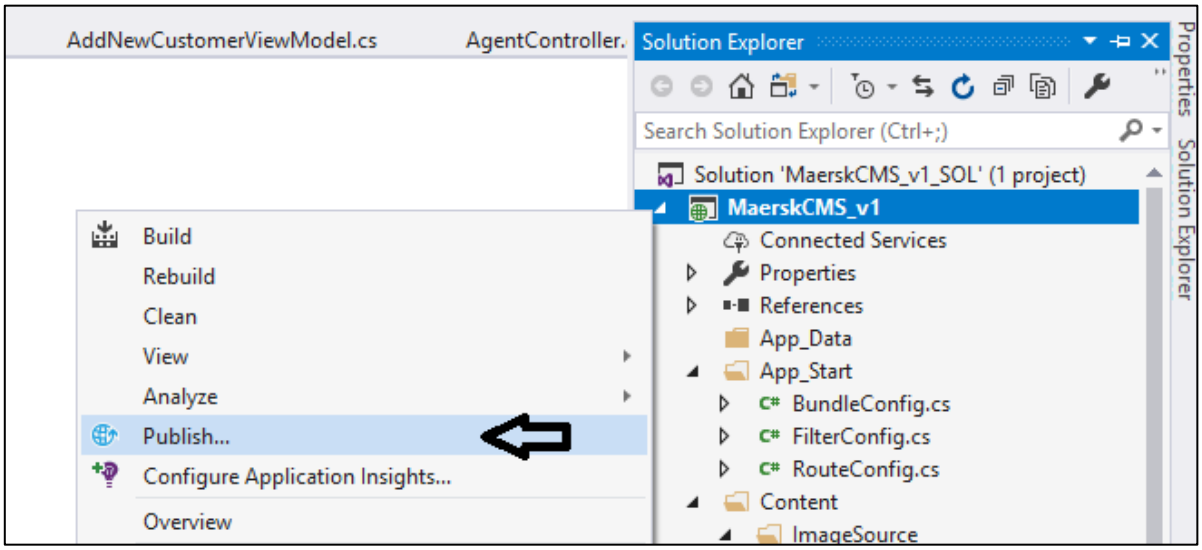


Figure 4-15 - Opening the publish option through the solution explorer in the Visual Studio IDE.

Next, the developer selects the target to be published by logging in the Azure account, selecting appropriate subscription, and select the target App Service available in the MaerskCMS_Group which is created in the section 4.2.2.

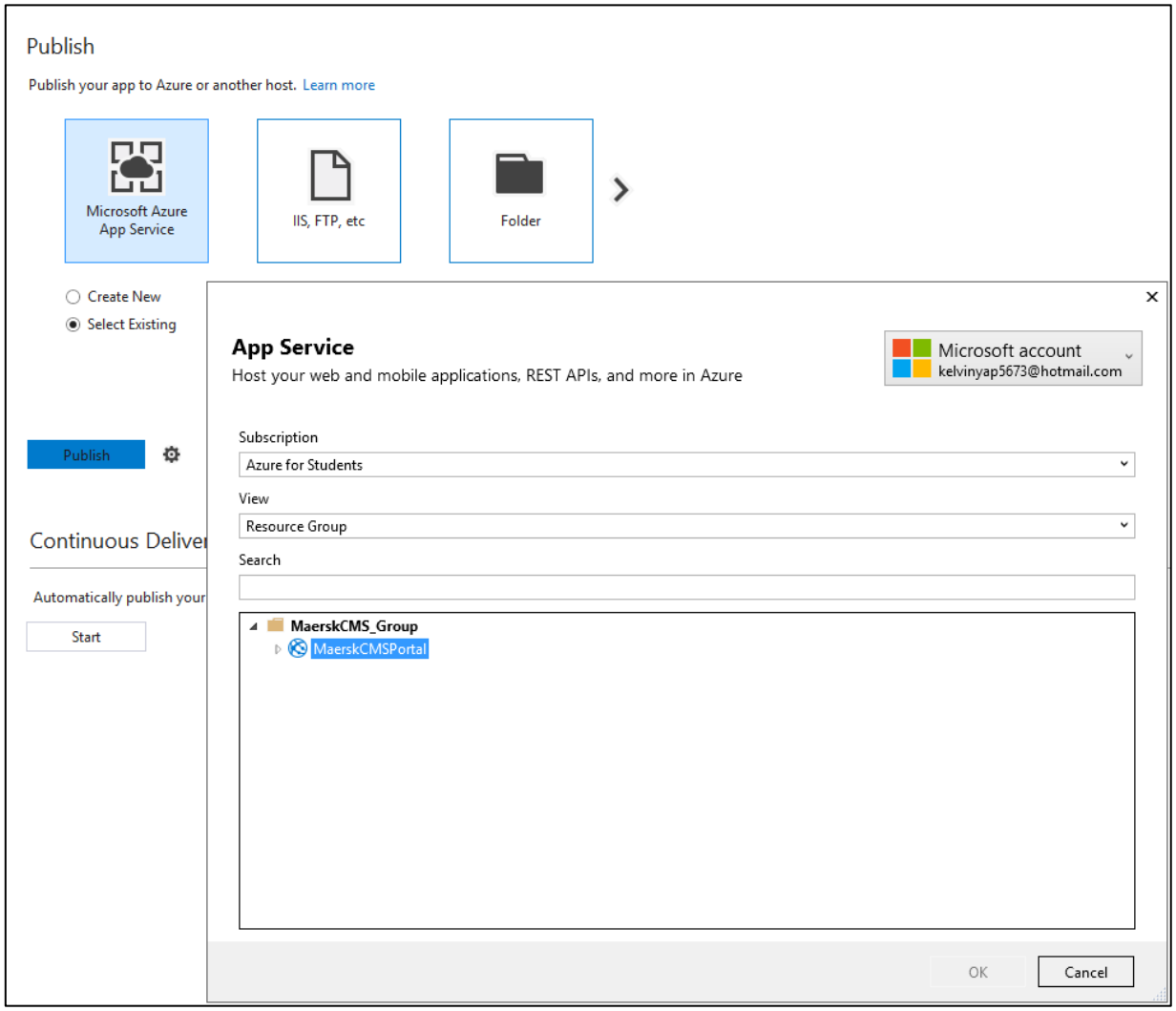


Figure 4-16 - Publishing the Application to the Azure App Service

4.2.5 Connecting the Azure Database

In this step, the developer connects the published Web Application and the SQL Database together by getting the connection strings from the Azure Portal and pasting the connection strings into the module located in the Web.Config file. This allows the Web Application to be granted access the SQL Database in Azure.

4.2.6 Traffic Manager (Optional)

This step is an optional step that allows the extension of the application performance to host the application on two or more servers with different profiles so that the user will be directed to the nearest server for improving web application response.

4.3 APPLICATION SCALING

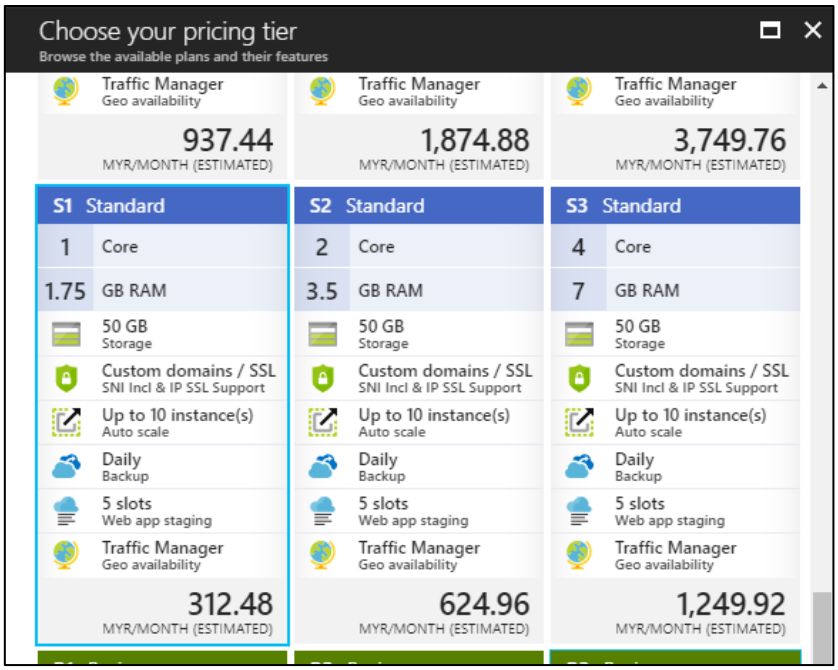


Figure 4-17 - The Chosen Pricing Tier of the App Service Plan

To be able to handle the high usage on peak hours with effective cost, the developer has decided to select the S1 Standard Tier for the best price for performance. The reason of the choice is because it has the best price for the essential features to run the Maersk Line CMS such as custom domains. This tier allows for up to 10 instances to be created and auto scaling is enabled based on metrics such as server load, memory usage, or even on a set schedule (Luijbregts, 2017). The standard tier also features daily backup of the configuration and the server data to ensure no data is lost when catastrophic event occurs. For the CI/CD workflow employed, this is needed. The deployment slots also allow the application to be rolled back to the last-functional slot in event that a security breach or bug is discovered in a build already rolled out to production.

Furthermore, the minimum requirement of the tier where the Traffic Manager is supported is the standard tier. The Traffic Manager allows user to access the web application through the nearest server to their region in favour of web application response.

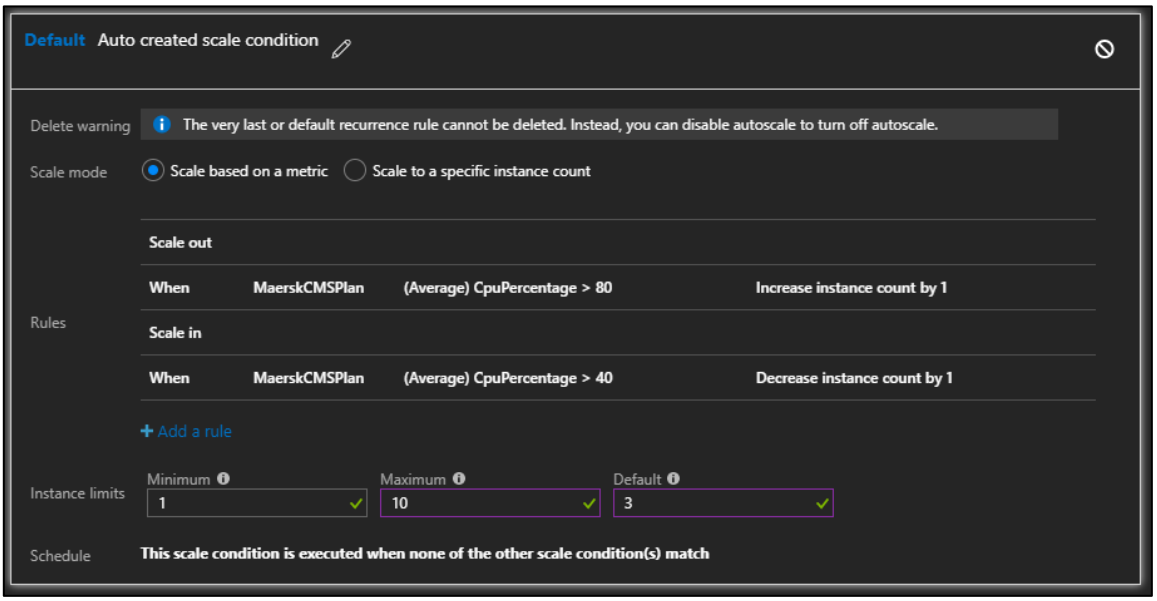


Figure 4-18 - Configuring the auto scale

In the auto scale configuration, the default instances count will be 3, minimum will be 1 and maximum will be 10. Additional instance will be added whenever the server CPU utilization increases above 80% and an instance will be removed when it drops below 40% In the future when business needs expands, the application service can be scaled to use the premium or isolated plans instead which provides even more features such as higher backup frequencies and better hardware such as SSDs (Lin, 2016).

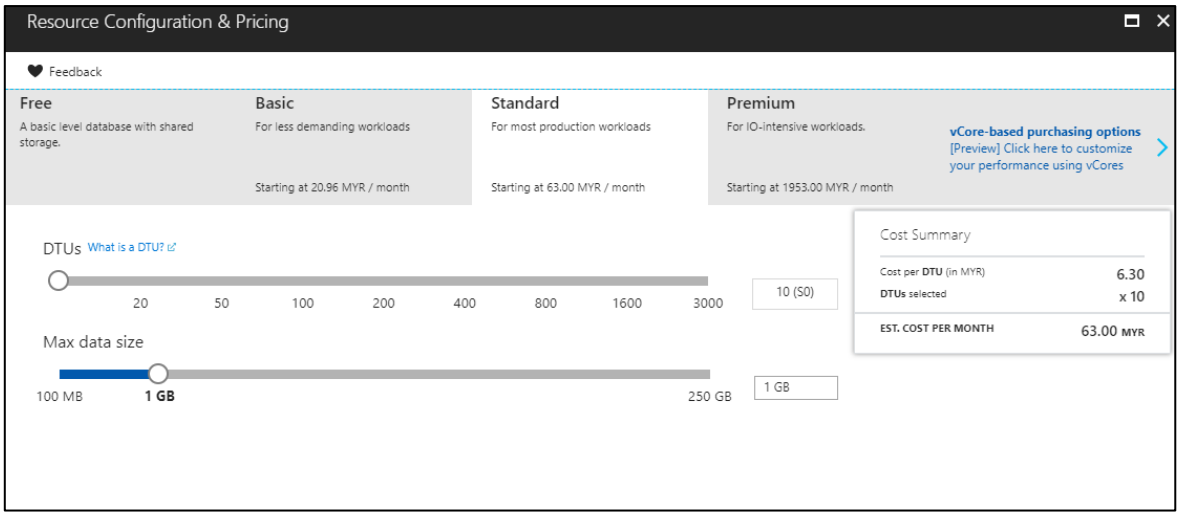


Figure 4-19 - SQL Database configuration and pricing.

At the initial deployment of the Maersk Line CMS, the developer expects low transactions performed in the first month. So, the developer has chosen 10 compute units for computation and 1GB for storage size which is sufficient enough for testing the application and the first month of the release. Upon the business growth, the configuration can be scaled up to handle the greater amount of transaction and storage.

4.4 RELIABILITY & PERFORMANCE

First of all, the reliability improvement is done through deploying the application on two or more servers in different regions as mentioned above. In this case, developer has deployed the application in Southeast Asia. This can improve the response time and throughout, so user would have the greatest experience when browsing Maersk Line CMS website with the help of traffic manager. Besides that, the web application will be assigned with 3 instances for high reliability which could scaled out to 10 instances. The resource allocation could be done effectively by Azure with load balancer to distribute the task among all the instances. The auto scaling is applied here as well to handle unexpected work load.

5. TESTING

5.1 FUNCTIONAL TESTING

Case No.	Unit	Test Description	Expected Result	Result
1	Agent Login	When in Agent Login Page, enter username & password, then click on Login Button	If username and/or password is invalid, error message will prompt. If entered credential is correct, web server redirect to Agent main page	Successful
2	Admin Login	When in Admin Login Page, enter username & password, then click on Login Button	If username and/or password is invalid, error message will prompt. If entered credential is correct, web server redirect to Admin main page	Successful
3	Logout	Click on Logout Link on the navbar	User will log out and redirect to home page	Successful
4	Agent Register	Fill up the required information and click Sign Up button	If all field is entered correctly, the web server will register the account and redirect to success page If any field is empty, or username is taken, or password is not valid, or Confirm Password is not valid, error message prompt.	Successful
5	Manage Customers	Login as Agent, click on Manage Customers in Navigation Bar	The web server returns the records of the customers if available	Successful
6	Delete Customer	Click on the Delete link on the desired customer record	The web server deletes the chosen customer record and refresh the page	Successful
7	Add New Customer	Fill up the required information and click Create button	If invalid field found, error message will prompt, otherwise, web server saves the information to the database server and redirect user to Manage Customers page	Successful
8	Edit Customer	Fill up the required information and click Edit button	If invalid field found, error message will prompt, otherwise, web server updates the information to the database server and redirect user to Manage Customers page	Successful

9	Manage Shipments	Login as Agent, click on Manage Shipments in Navigation Bar	The web server returns the records of the associated shipments if available	Successful
10	Delete Shipment	Click on the Delete link on the desired shipment record	The web server deletes the chosen shipment record and refresh the page	Successful
11	View Shipment Details	Click on the Details link on the desired shipment record	The web server redirect user to the chosen shipment record to view the details of the shipment.	Successful
12	View Schedule	Login as Agent, click on Shipping Schedule in Navigation Bar	The web server returns the records of the shipping schedule if available	Successful
13	Book New Shipment	User select an available shipping schedule, then fill in the required information and click on Submit.	If invalid field found, error message will prompt, otherwise, web server updates the information to the database server and redirect user to Agent Main page	Successful
14	Manage Vessel	Login as Admin, click on Manage Vessel in Navigation Bar	The web server returns the records of the vessels if available	Successful
15	Add New Vessel	Fill up the required information and click Create button	If invalid field found, error message will prompt, otherwise, web server saves the information to the database server and redirect user to Manage Vessel page	Successful
16	Delete Vessel	Click on the Delete link on the desired vessel record	The web server deletes the chosen vessel record and refresh the page	Successful
17	Manage Agents	Login as Admin, click on Manage Agents in Navigation Bar	The web server returns the records of the agents and the pending approval registrations if available	Successful
18	Approve Registration	Click on Approve link in the New Registration to be Approved table on the desired agents.	The web server updates the registration status of the chosen agents and refresh the page	Successful
19	Reject Registration	Click on Reject link in the New Registration to be Approved table on the desired agents.	The web server deletes the chosen agent record and refresh the page	Successful
20	Delete Agent	Click on Delete link in the Agent List table on the desired agents.	The web server deletes the chosen agent record and refresh the page	Successful

21	Manage Shipping Schedule	Login as Admin, click on Manage Shipping Schedule in Navigation Bar	The web server returns the records of the shipping schedule if available	Successful
22	Add New Shipping Schedule	Fill up the required information and click Create button	If invalid field found, error message will prompt, otherwise, web server saves the information to the database server and redirect user to Manage Shipping Schedule page	Successful
23	Delete Shipping Schedule	Click on the Delete link on the desired shipping schedule record	The web server deletes the chosen shipping schedule record and refresh the page	Successful

5.2 PERFORMANCE TESTING

The performance of the Maersk Line CMS can be measured by conducing a series of performance test which is to simulate the number of user visiting the site concurrently within a time limit. This feature can be used if the developer had selected the Standard Tier and above of the Service Plan. The test is conducted only within the main application in the main resource group which is MaerskCMS_Group by testing 500 users, 750 users and 1000 users visiting the site concurrently in 2 minutes of duration.

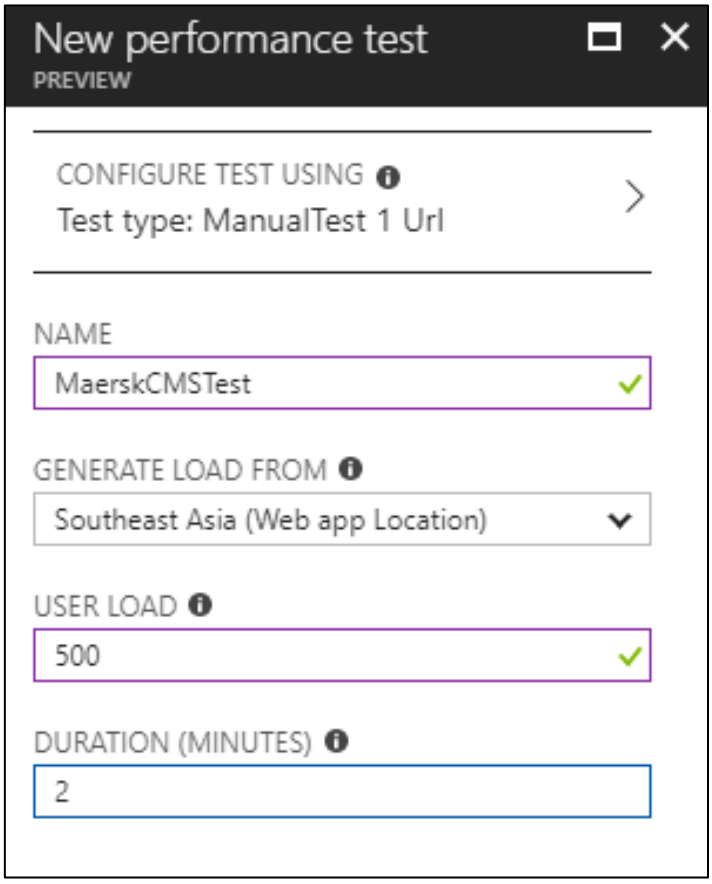


Table 5-1 - Creating a new performance test profile.

Starting with the Service Plan of S1 Standard Tier, the result of the performance testing is shown as the pie chart of successful and failure requests, status message, performance under load, CPU time, and memory usage together with the application scaling operation. However, the developer will only be focusing on the number of request and response time of the site to determine the performance of the Web App.

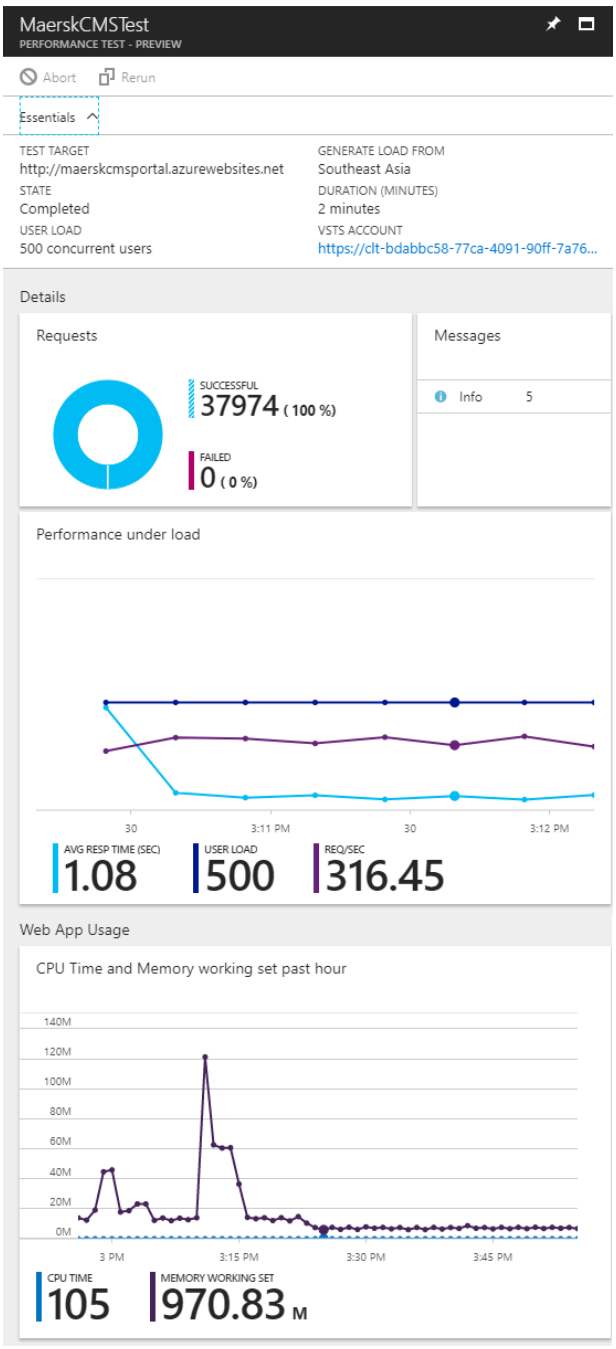


Table 5-2 - Testing 500 Users in 2 Minutes

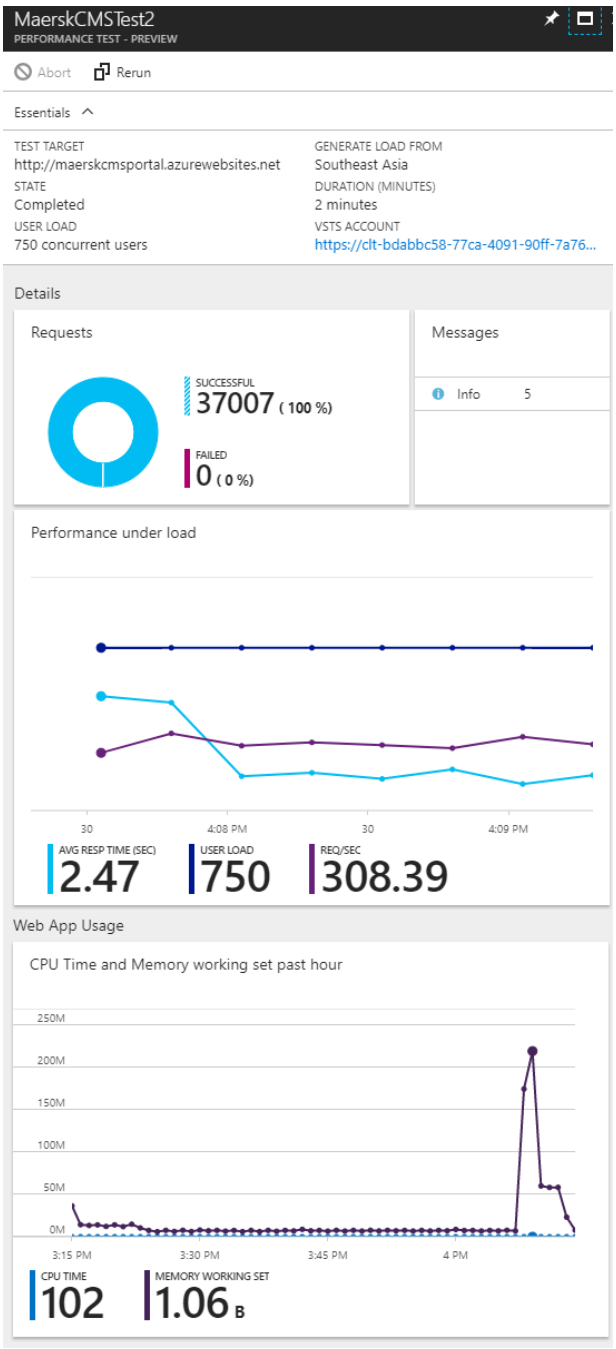


Table 5-3 - Testing 750 Users in 2 Minutes

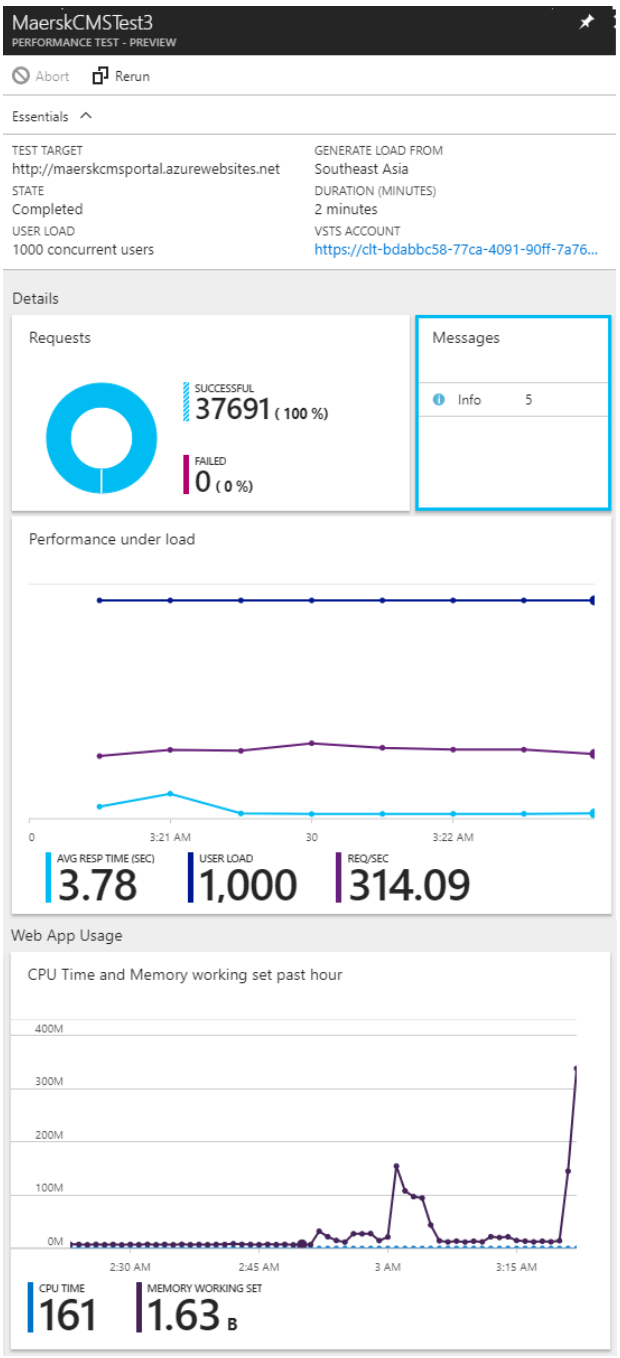


Table 5-4 - Testing 1000 Users in 2 Minutes

Based on the results above, the developer has confirmed that the current chosen tier of the Service Plan is sufficient of serve more than 1000 users concurrently as there are no failure / downtime of the Web App presents. Still, the upscaling can also be performed if the future growth of the business is obvious to overcome the performance issues.

To further investigate the impact of selecting different Standard Tier of the Service Plan, the developer has conducted more performance testing with using the Standard Tier S1, S2 and S3 App Service plan with the same number of users which are 500, 750 and 1000 users. The table shows the result of the testing.

Concurrent Users / **App Service Plan**	**500**	**750**	**1000**
S1	1.08 seconds 0 Fails	2.47 seconds 0 Fails	3.78 seconds 0 Fails
S2	0.56 seconds 0 Fails	0.78 seconds 0 Fails	1.10 seconds 0 Fails
S3	0.43 seconds 0 Fails	0.59 seconds 0 Fails	0.89 seconds 0 Fails

The conclusion can be made here is that higher tier has the lowest response time to response the requests from swarm of users. Although S3 has outperformed the Tier S1 and S2, but due to the high pricing, S3 is not the best choice for Maersk Line CMS. All in all, the Tier S2 is the most suitable choice for deploying the Web App with efficient cost.

6. MANAGED DATABASES

Platform as a Service (PaaS), is a category of cloud computing that provides a platform and an environment to allow developers to build applications and services over the Internet by accessing via the web browser (Interoute Communications Limited, 2018). With the increasing popularity of cloud and managed services available online, more and more services previously only available through a locally provisioned server are available at a click of a button. PaaS has become a valuable and profitable tool to various market which desired to speed up the development and deployment process. The PaaS provider such as Microsoft, Google, Amazon, and Red Hat provides many PaaS such as Microsoft Azure, Google App Engine, Amazon AWS, Red Hat OpenShift and many more (Sullivan, 2014).

There are lots of benefits of using a PaaS than purchase a physical server, an Internet package, or hire professionals to maintain the server in terms of cost and time.

- Scalability: The rapid allocation and deallocation of resources with a pay-as-you-use model. For example, the developer can enable auto-scaling in Microsoft Azure services to ensure only the required resources are used when required.
- Reduction of capital expenditure: Same as above, by having the auto-scaling feature, the cost only consists of the resources used when needed.
- Reduction of lead times with on-demand availability of resources: For example, the developer is able to create 10 servers with a single script, then remove it after 5 minutes of testing.
- Self-service with reduced administration costs: The developer who use the PaaS is able to control all the functionalities without any external help.
- Reduced skill requirements: Security is one of the important features in PaaS which the developer does not required to have the skill to overcome.
- Support of team collaboration: Resource group can be shared to multiple development teams, avoiding the need for wasteful allocation of multiple assets of the sae type in separate sites.
- Easier and quicker user access control: The resource group can be easily grant access to the user on a few clicks of the buttons.

(Meegan, 2016)

Furthermore, most of the PaaS such as Amazon AWS guarantee the availability of the service is 99% of all time, which further increase the reliability of the deployed services at the PaaS to significantly mitigate the downtime of a service (Amazon Web Services, 2013).

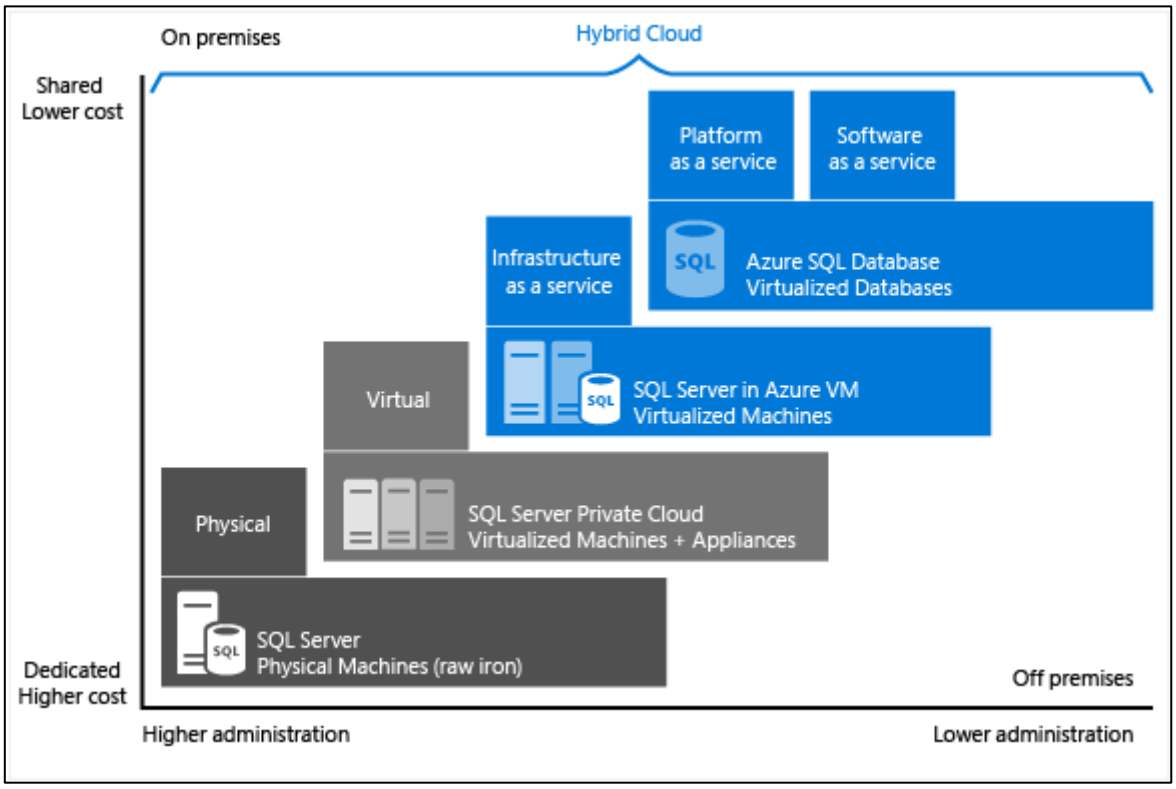


Figure 6-1 - The characteristic of level of administration over degree of cost efficiency

The most interesting part of PaaS provided is the managed database service. According to Microsoft (2018), SQL Server on Microsoft Azure virtual machines enables the company to meet unique and diverse business needs with a combination of on-premises and cloud-hosted deployments, while using the same set of server products, development tools, and expertise across the physical on-premises machines, private cloud environments, third party hosted private cloud environments, and public cloud. From the diagram above, each offering can be characterized by the level of administration that developer has over the infrastructure, and by the degree of cost efficiency achieved by database level consolidation and automation (Microsoft, 2018). This means that the PaaS model allows lower administration but advanced features at lower costs by introducing built-in functionalities that covers most of the requirements of deploying an SQL database (Microsoft, 2018). The reduction the overall costs to spend on provisioning and managing the database server can be foreseen through the use of the PaaS (Microsoft, 2018). Besides, the reduction of the needs for ongoing administration is present as Microsoft Azure has well-managed the virtual machines, operating system and database software to provide the best possible experience for the developer as well as the novice user of the PaaS (Microsoft, 2018).

On the contrary, the downside of using the PaaS is that the data is stored in the servers of a third-party providers, posing the risk of data leakage to the providers if cryptography is not used (AIA, 2015). The lack of access to the physical infrastructure of the machine is also a downside of using PaaS. There is

a risk present in customizing or developing an application in a cloud environment of a vendor which will not migrate easily to another environment, mostly happens with the use of an open source PaaS (IDG Communications, Inc., 2018). Moreover, the software compatibility may be an issue for some enterprise which use unstandardized software, programming languages or tools as most of the PaaS supports the popular and standardized software, programming languages or tools (IDG Communications, Inc., 2018). A native support from the vendor is crucial to minimize the complexity of the infrastructure, maximize the performance and scalability (IDG Communications, Inc., 2018).

7. CONCLUSION

In conclusion, the Maersk Line CMS project was successfully developed and completed with deployment on the Azure cloud environment. Throughout the project lifetime, the developer has gained a fundamental understanding on cloud computing in different forms. More than that, the developer has acknowledged the use of the Microsoft Azure and the potential of using the Microsoft Azure for the expertise and enterprise usage. The developer has learned the configuration of the Microsoft Azure cloud environment which the Maersk Line CMS would be deployed to. Furthermore, the developer also understands the ways to develop and deploy the Web Application or Services based on the project specification. Last but not least, the developer believes that gaining the knowledge about the cloud computing can boost the development skills of the developer for proceeding the career path since the cloud computing has become a hit these days.

8. REFERENCE

Luijbregts, B. (2017). *How to Autoscale Azure App Services & Cloud Services.* [Online]. Available from: https://stackify.com/autoscale-azure-app-services-cloud-services/. [Accessed: 15th April 2018]

Lin, C. (2016). *Scale up an app in Azure.* [Online]. Available from: https://docs.microsoft.com/en-us/azure/app-service/web-sites-scale. [Accessed: 15th April 2018]

Interoute Communications Limited. (2018). *What is PaaS?* [Online]. Available from: https://www.interoute.com/what-paas. [Accessed: 18th April 2018]

Sullivan, D. (2014). *PaaS Providers List: Comparison And Guide.* [Online]. Available from: http://www.tomsitpro.com/articles/paas-providers,1-1517.html. [Accessed: 18th April 2018]

Meegan, J. (2016). *A practical guide to platform as a service: PaaS benefits and characteristics.* [Online]. Available from: https://www.ibm.com/blogs/cloud-computing/2016/08/22/paas-benefits-characteristics/. [Accessed: 18th April 2018]

Microsoft. (2018). *Choose a cloud SQL Server option: Azure SQL (PaaS) Database or SQL Server on Azure VMs (IaaS).* [Online]. Available from: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-paas-vs-sql-server-iaas. [Accessed: 18th April 2018]

IDG Communications, Inc. (2018). *PaaS: 3 Critical Issues to Consider.* [Online]. Available from: https://www.cio.com/article/2939697/cloud-computing/paas-3-critical-issues-to-consider.html. [Accessed: 19th April 2018]

AIA. (2015). *Advantages and Disadvantages of SaaS, PaaS and IaaS.* [Online]. Available from: http://aiasecurity.com/2015/09/10/advantages-and-disadvantages-of-saaspaas-and-iaas/. [Accessed: 19th April 2018]