Asia Pacific University of Technology and Innovation

Design and Developing Application on the Cloud

CT071-3-3-DDAC

Project Title:	**Maersk Line Container Management System**
Intake Code:	**UC3F1706SE**
TP Number:	**TP037694**
Name:	**Tan Li Feng**
Lecturer Name:	**DR. KALAI ANAND A/L RATNAM**
Submission Date:	**13/04/2018**

Acknowledgement

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to our DDAC Lecturer, DR. KALAI ANAND A/L RATNAM, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report and every comment and advice. He had invested her full effort in guiding me in achieving the goal.

Table of Contents

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. 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.

Therefore, in this assignment, the resulting Maersk Line Online Container Management System will be developed and deployed onto the Azure cloud platform. With the advent of cloud services, it is no surprise that such a decision was made. The application provides Maersk Line's staffs the ability to manage shipping schedule and book vessel online.

1.2 Objective & Scope

Designing and developing a Container Management System (CMS) to cater to manage the containers, a solution that reduces overall supply chain costs and an efficient way to manage logistics.

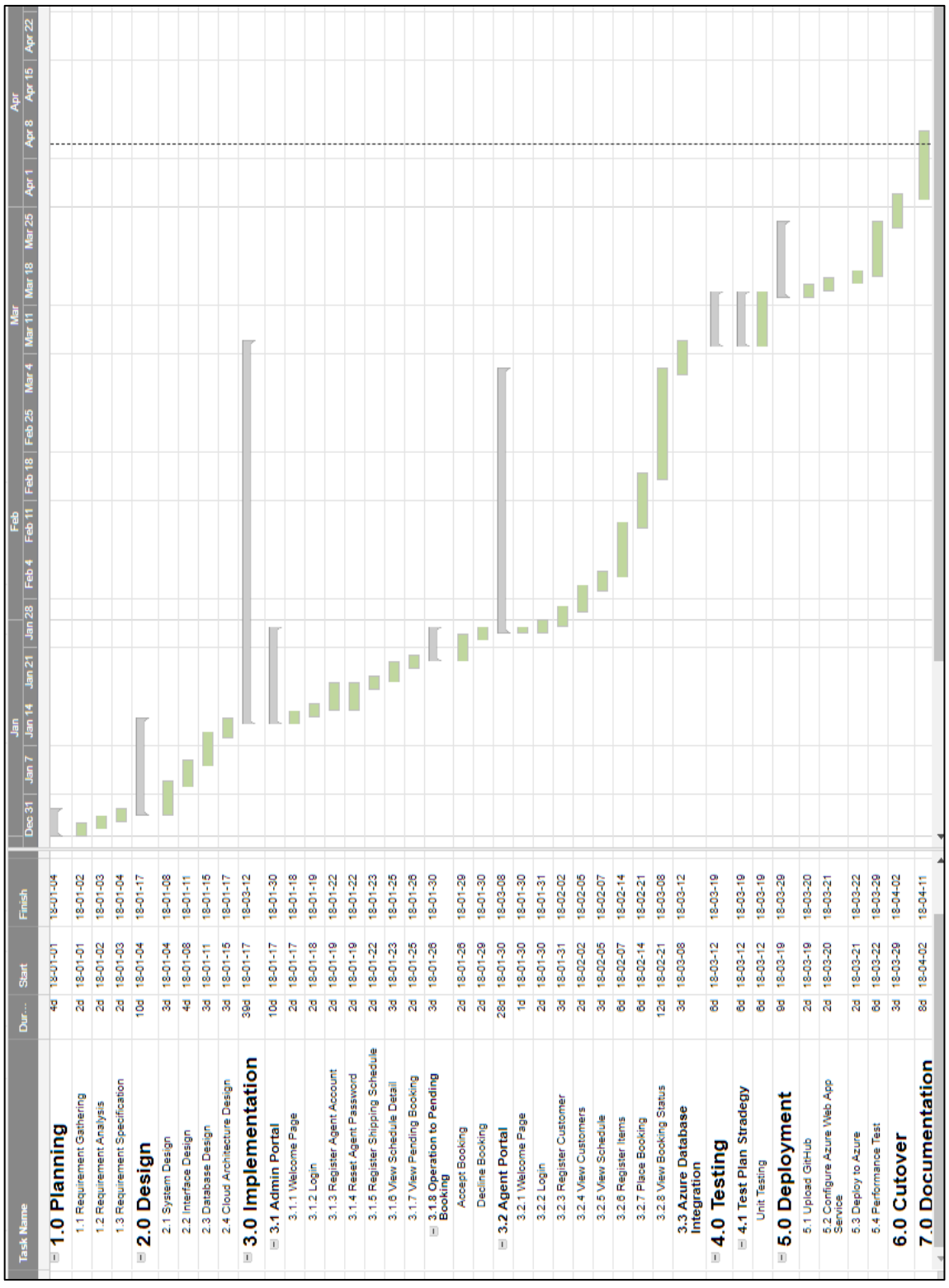
1.3 Requirements

1. From import, export and transshipment 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 entire booking process from schedule search to booking confirmation.

1.4 Deliverables & Fulfillment

1. Design & Develop a single tenant web application hosted on Microsoft Azure as an App Service (Web App) or on AWS Elastic Beanstalk.
2. Consume Relational Database
3. Consist of 5 - 10 interlinked pages.
4. Provide quality content and design.
5. Analyze 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.

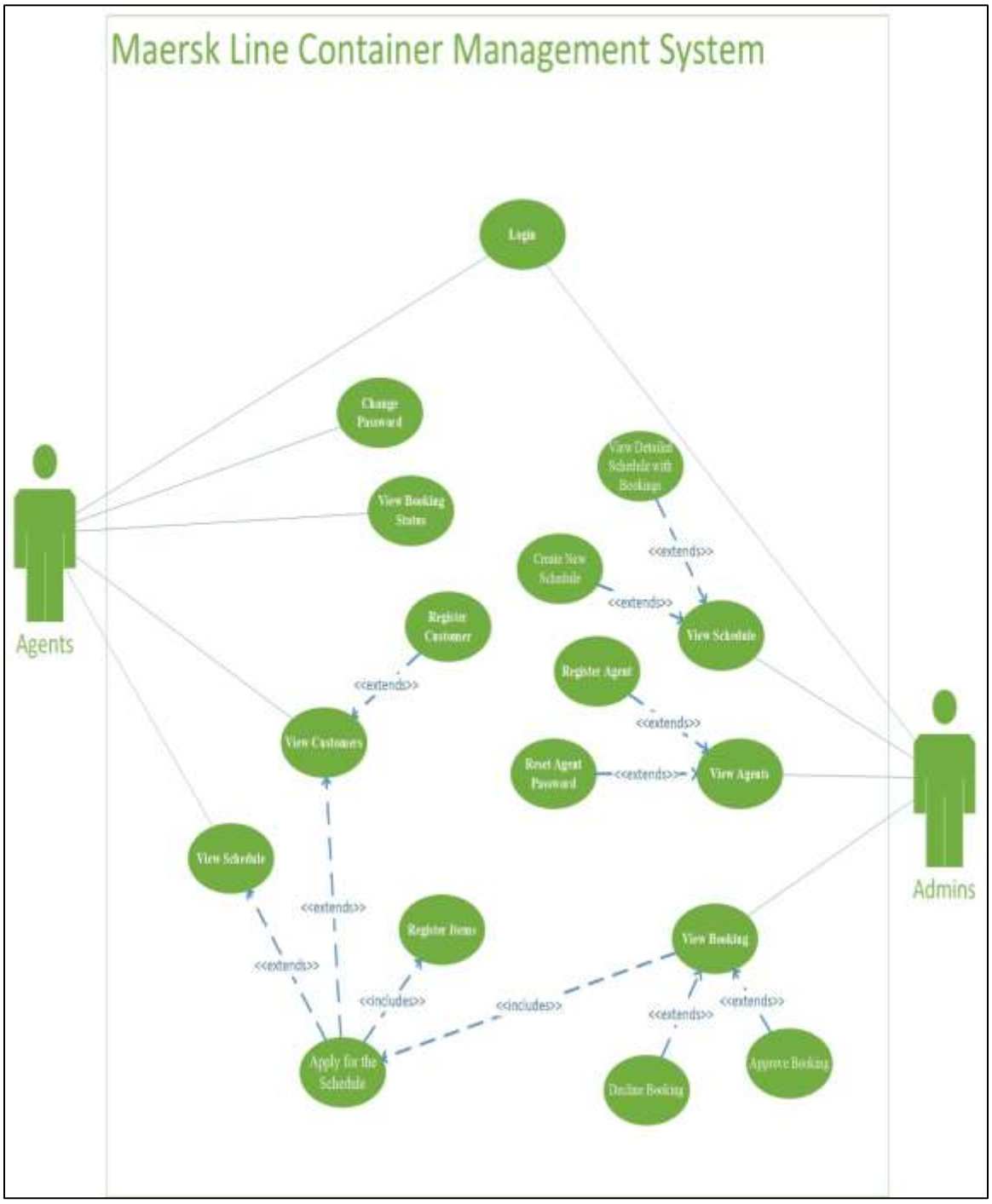
2.0 Project Plan



Task Name	Dur...	Start	Finish
1.0 Planning	4d	18-01-01	18-01-04
1.1 Requirement Gathering	2d	18-01-01	18-01-02
1.2 Requirement Analysis	2d	18-01-02	18-01-03
1.3 Requirement Specification	2d	18-01-03	18-01-04
2.0 Design	10d	18-01-04	18-01-17
2.1 System Design	3d	18-01-04	18-01-08
2.2 Interface Design	4d	18-01-08	18-01-11
2.3 Database Design	3d	18-01-11	18-01-15
2.4 Cloud Architecture Design	3d	18-01-15	18-01-17
3.0 Implementation	39d	18-01-17	18-03-12
3.1 Admin Portal	10d	18-01-17	18-01-30
3.1.1 Welcome Page	2d	18-01-17	18-01-18
3.1.2 Login	2d	18-01-18	18-01-19
3.1.3 Register Agent Account	2d	18-01-19	18-01-22
3.1.4 Reset Agent Password	2d	18-01-19	18-01-22
3.1.5 Register Shipping Schedule	2d	18-01-22	18-01-23
3.1.6 View Schedule Detail	3d	18-01-23	18-01-25
3.1.7 View Pending Booking	2d	18-01-25	18-01-26
3.1.8 Operation to Pending Booking	3d	18-01-26	18-01-30
Accept Booking	2d	18-01-26	18-01-29
Decline Booking	2d	18-01-29	18-01-30
3.2 Agent Portal	28d	18-01-30	18-03-08
3.2.1 Welcome Page	1d	18-01-30	18-01-30
3.2.2 Login	2d	18-01-30	18-01-31
3.2.3 Register Customer	3d	18-01-31	18-02-02
3.2.4 View Customers	2d	18-02-02	18-02-05
3.2.5 View Schedule	3d	18-02-05	18-02-07
3.2.6 Register Items	6d	18-02-07	18-02-14
3.2.7 Place Booking	6d	18-02-14	18-02-21
3.2.8 View Booking Status	12d	18-02-21	18-03-08
3.3 Azure Database Integration	3d	18-03-08	18-03-12
4.0 Testing	6d	18-03-12	18-03-19
4.1 Test Plan Stradegy	6d	18-03-12	18-03-19
Unit Testing	6d	18-03-12	18-03-19
5.0 Deployment	6d	18-03-19	18-03-29
5.1 Upload GitHub	2d	18-03-19	18-03-20
5.2 Configure Azure Web App Service	2d	18-03-20	18-03-21
5.3 Deploy to Azure	2d	18-03-21	18-03-22
5.4 Performance Test	6d	18-03-22	18-03-29
6.0 Cutover	3d	18-03-29	18-04-02
7.0 Documentation	8d	18-04-02	18-04-11

3.0 Design

3.0 Use Case Diagram



3.1 Use Case Specifications

Use Case Name	Login
Summary	User is going to login to the system with given ID and Password.
Actor	Admin and Agent
Precondition	User has entered ID and Password
Main Sequence	1. System search for the ID in database and match with the entered password. 2. Matched successfully and show main menu to user according to their identity.
Alternative Sequence	1. If user enter wrong ID or Password, system unable to match anything in database, system will generate message to inform user.
Post condition	User log to the system successfully.

	AGENTS
Use Case Name	Change Password
Summary	Agent can change their password
Actor	Agent
Precondition	Agents are going to change their account password
Main Sequence	1. System will search for the employee's information within database by Agents ID. 2. Show the relevant information in GUI. 3. User replace old information in the editable field. 4. System updates the new information to the database.
Alternative Sequence	1. If user enter wrong data type, system will pop out a message box to inform user.
Post condition	System shows all the relevant information and update new information successfully

Use Case Name	Agents view schedule
Summary	Agents able to view the available shipping schedule
Actor	Agents
Precondition	Admin has created shipping schedules
Main Sequence	1. System will search for the available shipping schedule within database by Schedule ID. 2. System shows the details of shipping schedule.
Alternative Sequence	1. If admins didn't create any schedule, system will inform user.
Post condition	System shows the correct shipping schedule to the user.

Use Case Name	Agents view customers
Summary	Agents able to view the registered Customers
Actor	Agents
Precondition	Agent has created Customer
Main Sequence	1. System will search for the customers within database by Customer ID. 2. System shows the details of Customers.
Alternative Sequence	1. If agents didn't create any customer, system will inform user.
Post condition	System shows the correct customer to the user.

Use Case Name	Register Customer
Summary	Agents able to register New Customers
Actor	Agents
Precondition	User has no account of the system.
Main Sequence	1. Enter the customer information 2. System checks for availability of the entered IC. 3. If true, Account will be created.
Alternative Sequence	1. If user enter a unavailable IC, the system will generate a message box to inform the user.
Post condition	User registers the account successfully.

Use Case Name	Place Bookings
Summary	Agents able to place booking on a shipping schedule
Actor	Agents
Precondition	Admin has created shipping schedules
Main Sequence	1. Agents select the schedule. 2. Agents select the customer. 3. Insert Item Details 4. Press confirm button.
Alternative Sequence	1. If database does not has any schedule, the system will return a message to inform employees.
Post condition	System will notify the booking to Admins.

Use Case Name	Agents view booking status
Summary	Agents able to view the bookings and status.
Actor	Agents
Precondition	Agents placed several bookings via the system.
Main Sequence	1. System search for the bookings by the Booking ID. 2. Get and show the detail and status of the bookings
Alternative Sequence	1. If agent does not place any booking, the system will return a message to inform employees.
Post condition	System will show up the detail and status of the bookings.

	EMPLOYERS
Use Case Name	Admins view schedule
Summary	Admins able to view the available shipping schedule
Actor	Admins
Precondition	Admin has created shipping schedules
Main Sequence	1. System will search for the available shipping schedule within database by Schedule ID. 2. System shows the details of shipping schedule.
Alternative Sequence	1. If admins didn't create any schedule, system will inform user.
Post condition	System shows the correct shipping schedule to the user.

Use Case Name	View Agents
Summary	Admins able to view all the agents.
Actor	Admins
Precondition	Admin has created agents account
Main Sequence	1. System will search for the agents by Agents ID. 2. System shows all the agents
Alternative Sequence	1. If the database do not have agent accounts, the system will generate message to inform.
Post condition	System shows the correct details of agents.

Use Case Name	Register Agent
Summary	Admin able to register Agent Account
Actor	Admin
Precondition	User has no account of the system.
Main Sequence	1. User enter username, user details, password and confirmed password in the input box. 2. System checks for availability of the entered Username. 3. If true, Account will be created.
Alternative Sequence	1. If user enter a unavailable Username, the system will generate a message box to inform the user.
Post condition	User registers the account successfully.

Use Case Name	Reset Agent Account Password
Summary	Admin able to reset the password of agent account
Actor	Admin
Precondition	User has account in the system.
Main Sequence	1. Select the target agent 2. Click the reset password button 3. Password reset
Alternative Sequence	
Post condition	Agent account password is reseted successfully.

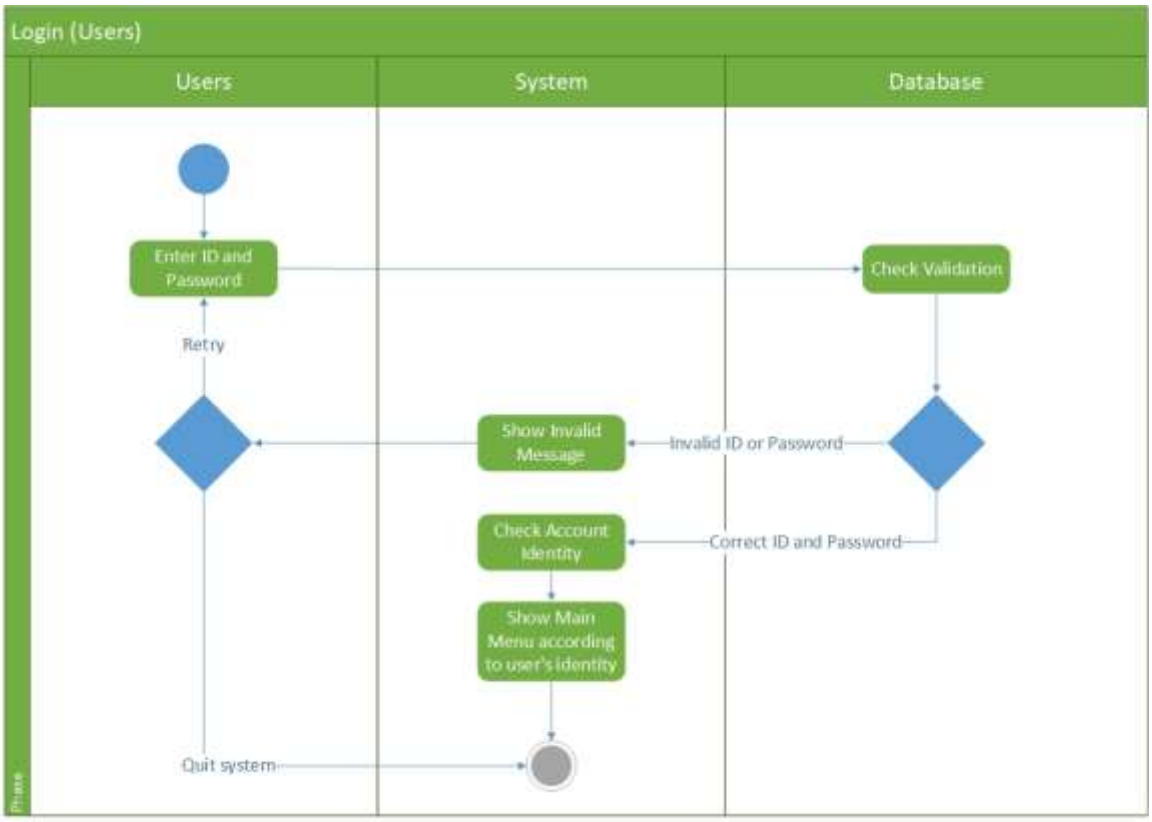
Use Case Name	View Pending Bookings
Summary	Admins able to view the list of the bookings and view the detail of customer and items.
Actor	Admins
Precondition	Agents had placed bookings.
Main Sequence	1. System search for the bookings by Booking ID 2. Matched successfully and show details of the item and customer. 3. Generate 2 button which are "Accept" and "Decline" on each booking row.
Alternative Sequence	1. If there are booking are placed, system will generate message to inform admins
Post condition	System shows bookings and provide either Accept or Decline operation for admins

Use Case Name	Accept or Decline Application
Summary	Admins able to either accept or decline bookings.
Actor	Admins
Precondition	Agents had placed bookings.
Main Sequence	1. Select either "Accept" or "Decline" the bookings from agent. 2. System will get the selected button and proceed either Accept or Decline process to the application. 3. Update to the database.
Alternative Sequence	1. If there are booking are placed, system will generate message to inform admins
Post condition	1. Accept - System marks the status of application to "Approved" and notifies Agents 2. Decline - System marks the status of application to "Declined" and notifies Agents

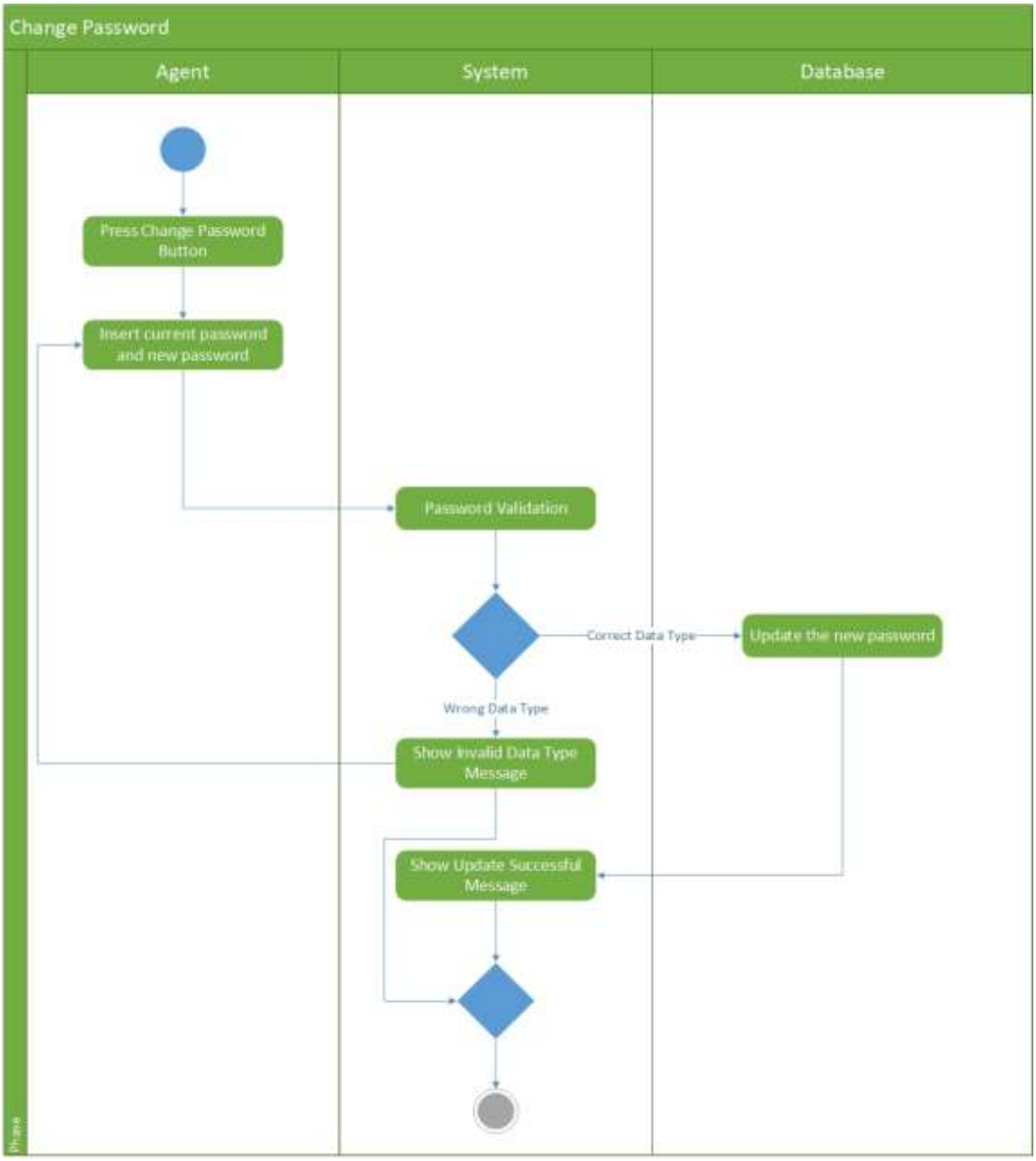
Use Case Name	Add New Schedule
Summary	Admins able to create new shipping schedule
Actor	Admins
Precondition	Login as System Admin
Main Sequence	1. Press Add Schedule Button 2. Enter the Shipping Information 3. Submit
Alternative Sequence	1. If user leaks enter information, the system will generate a message box to inform the user.
Post condition	Schedule is created successfully

3.2 Activity Diagram

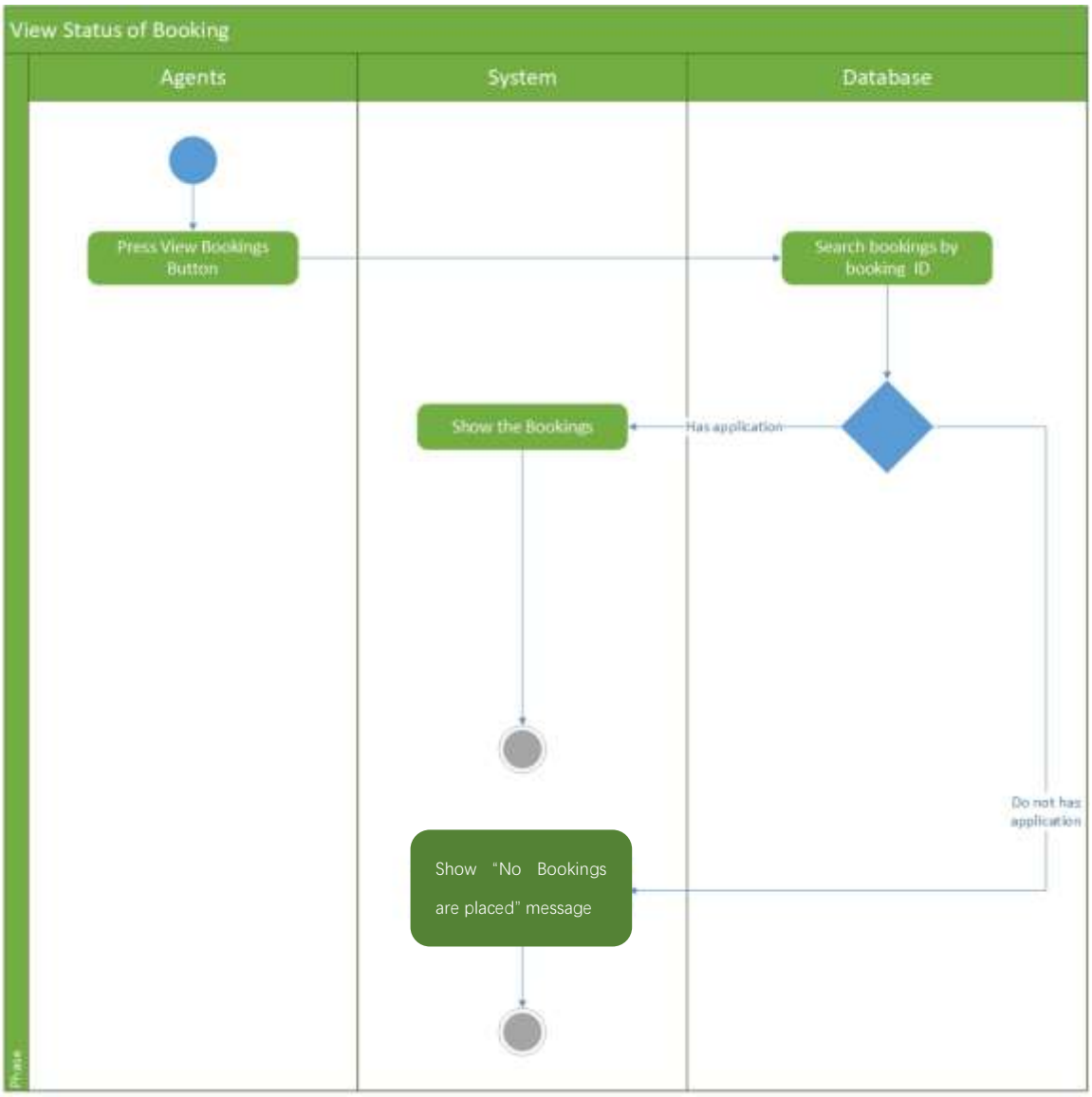
3.2.1 Login



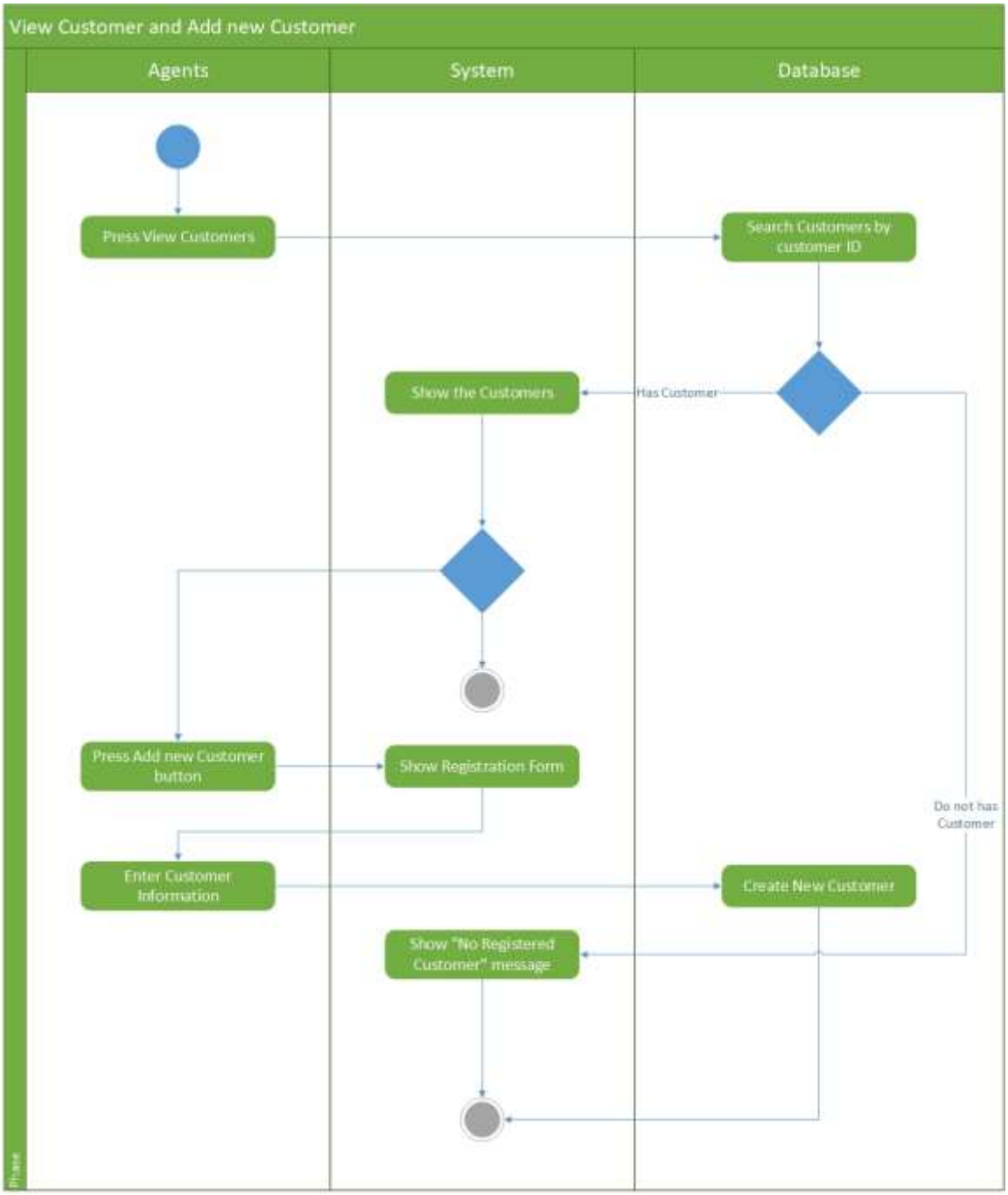
3.2.2 Change Password (Agents)



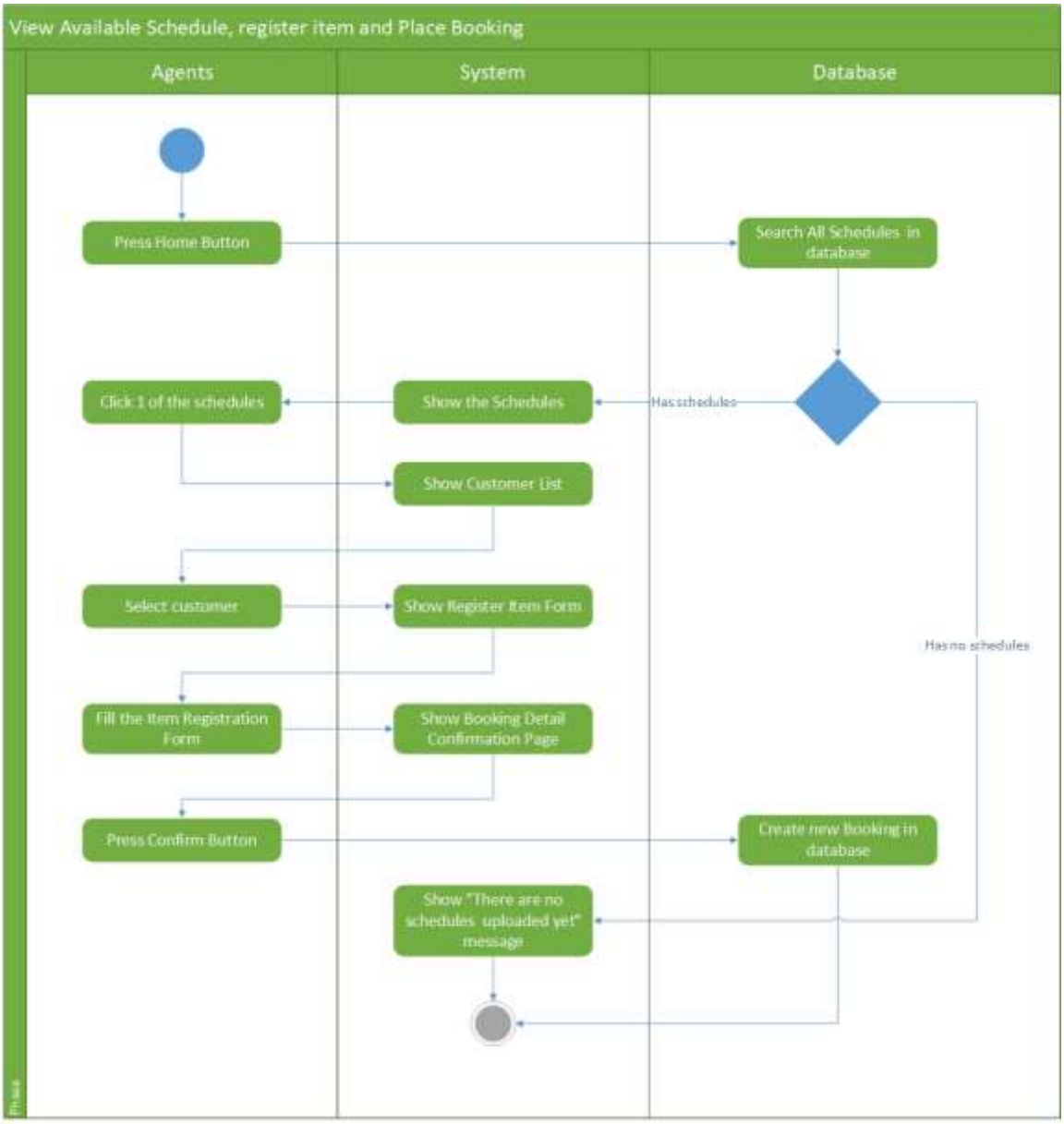
3.2.3 View Booking (Agents)



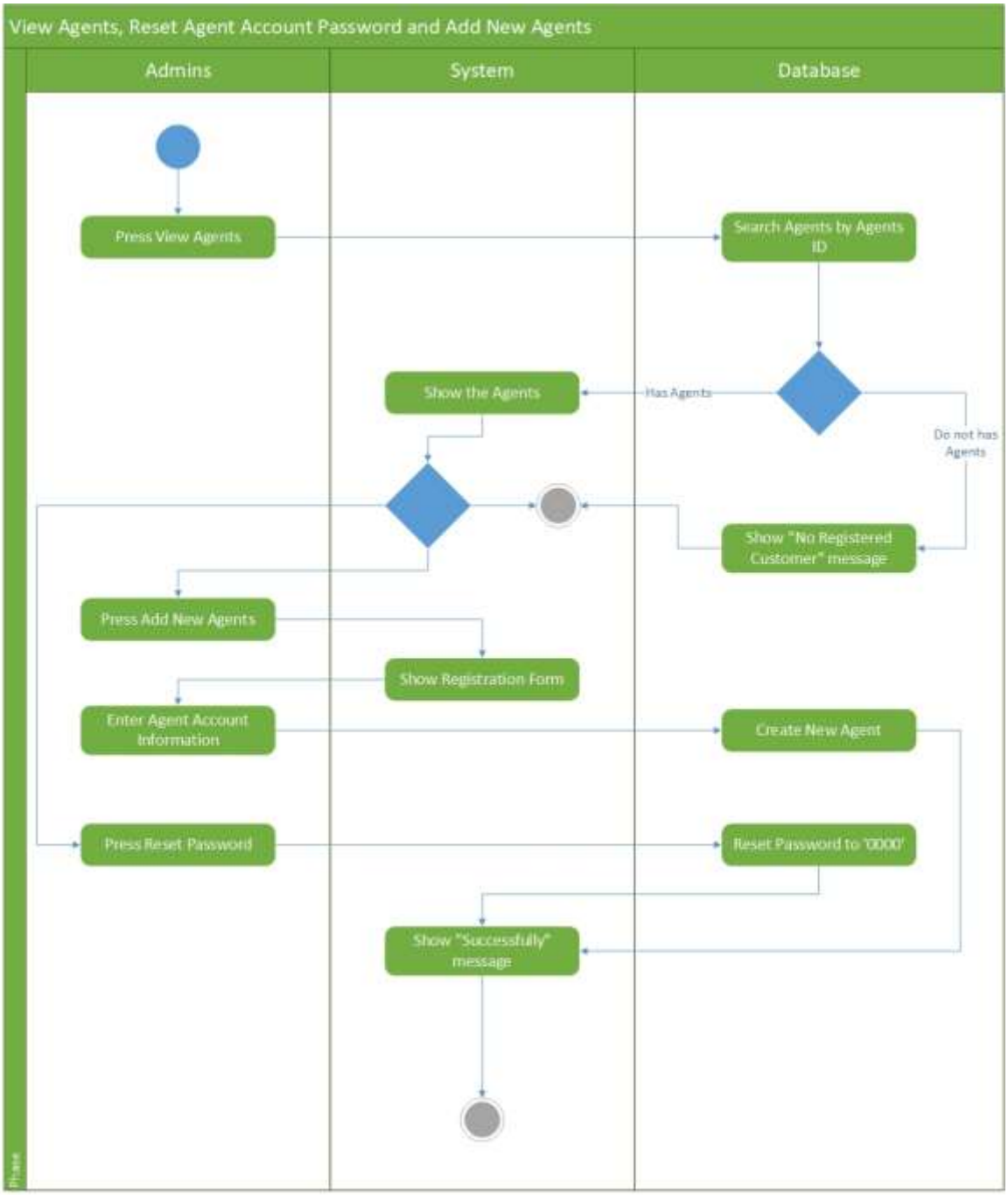
3.2.4 View Customer (Agents)



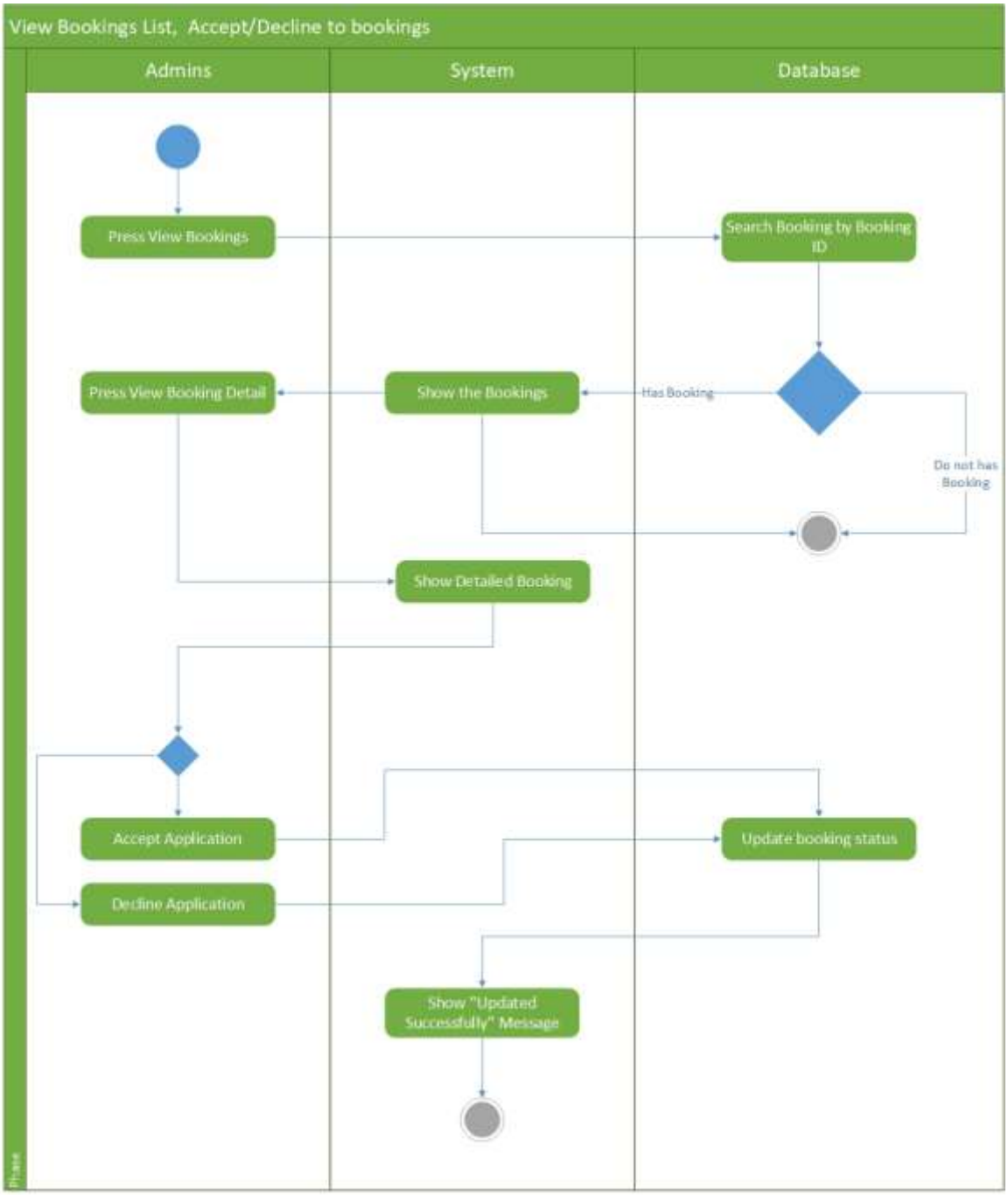
3.2.5 View Schedules (Agents)



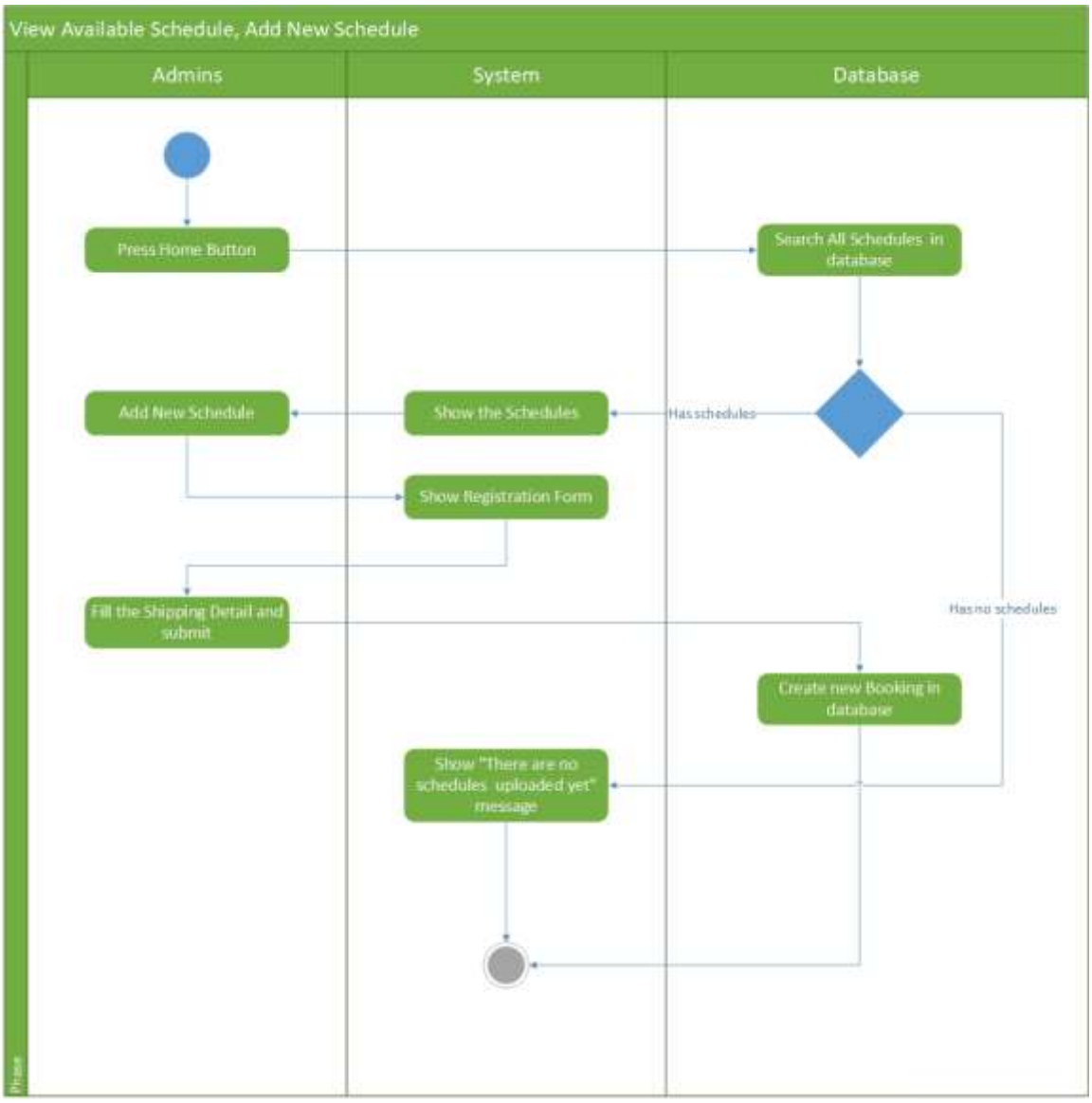
3.2.6 View Agents (Admins)



3.2.7 View Bookings (Admins)

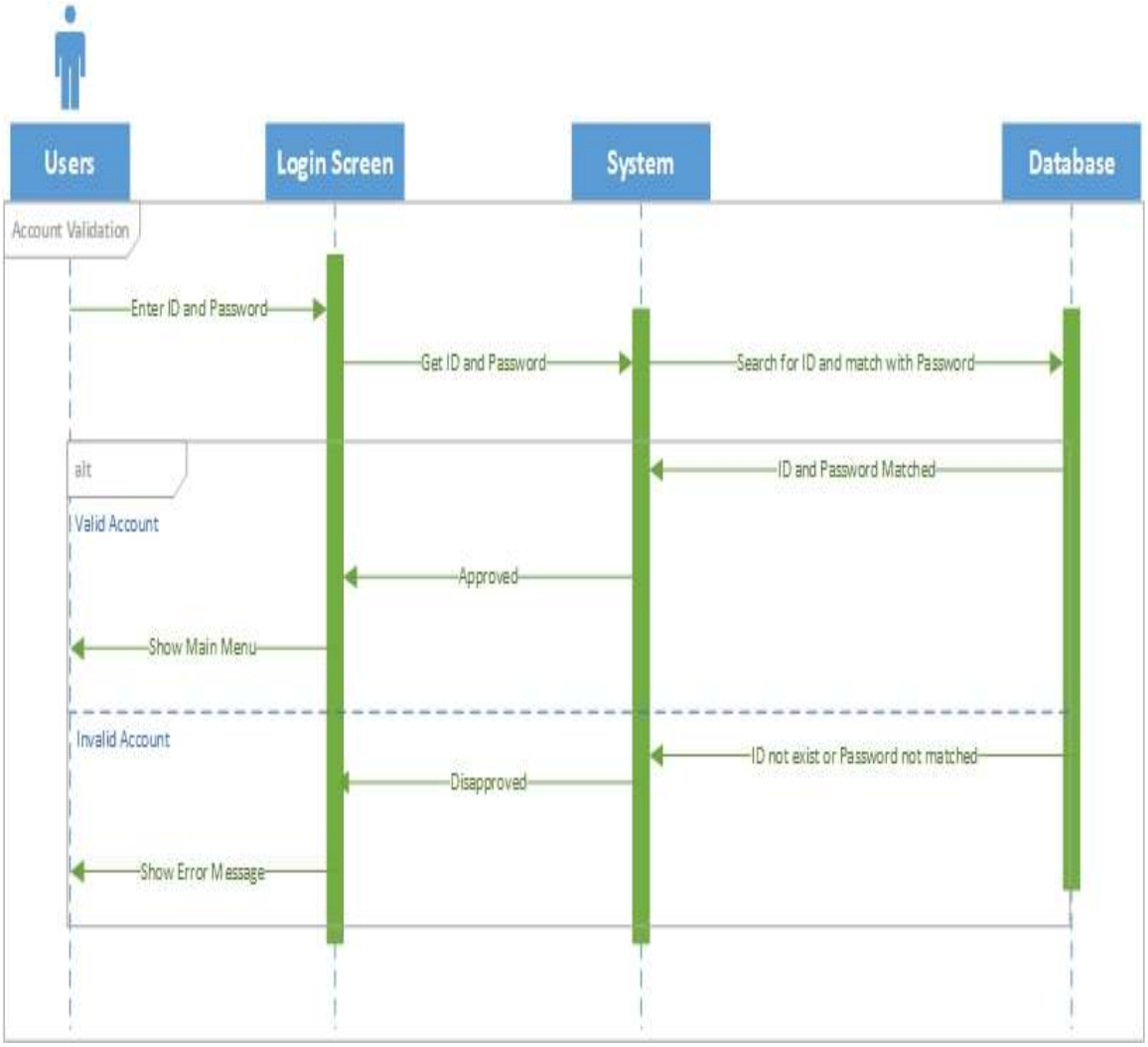


3.2.8 View Schedules (Admins)

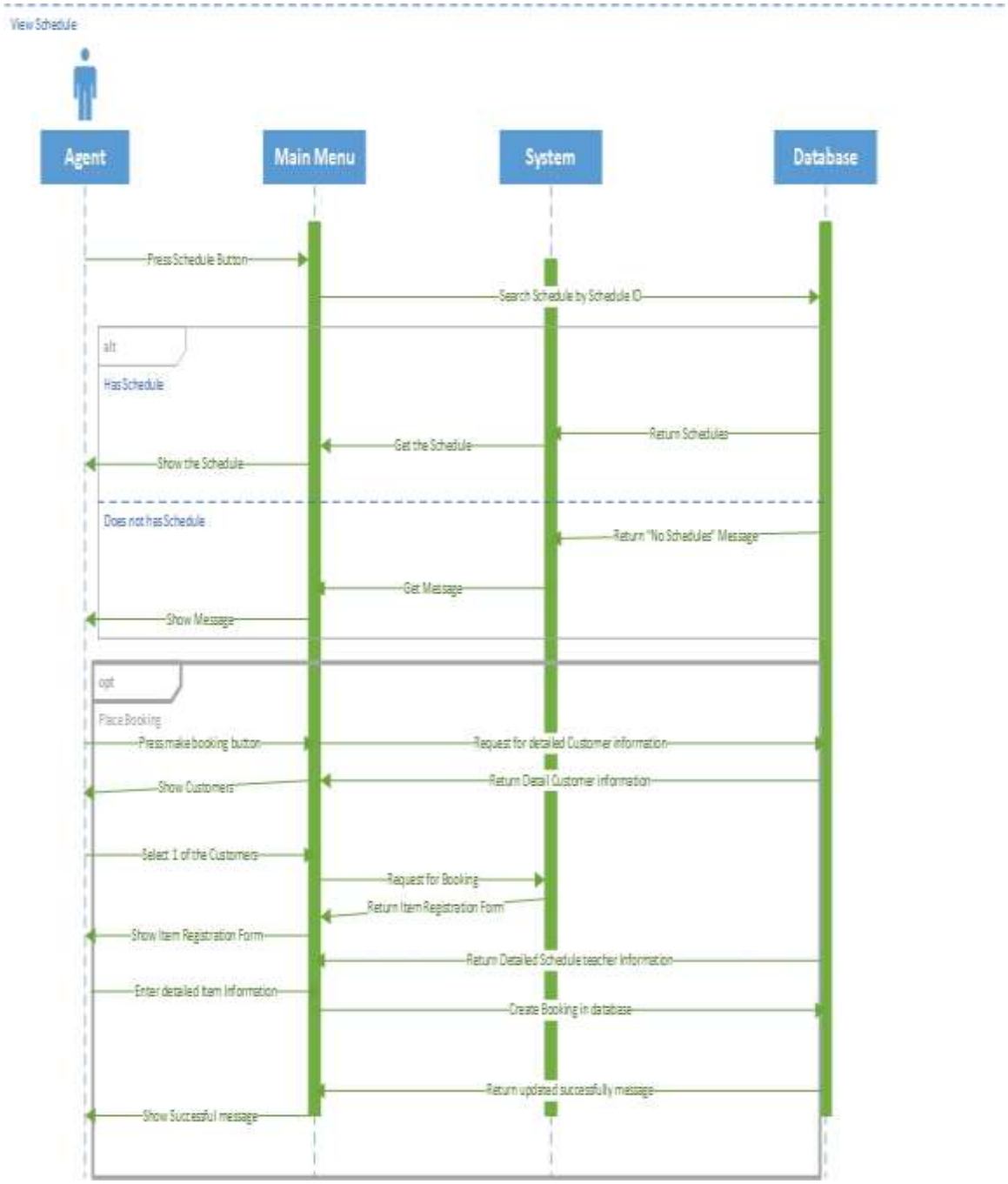


3.3 Sequence Diagram

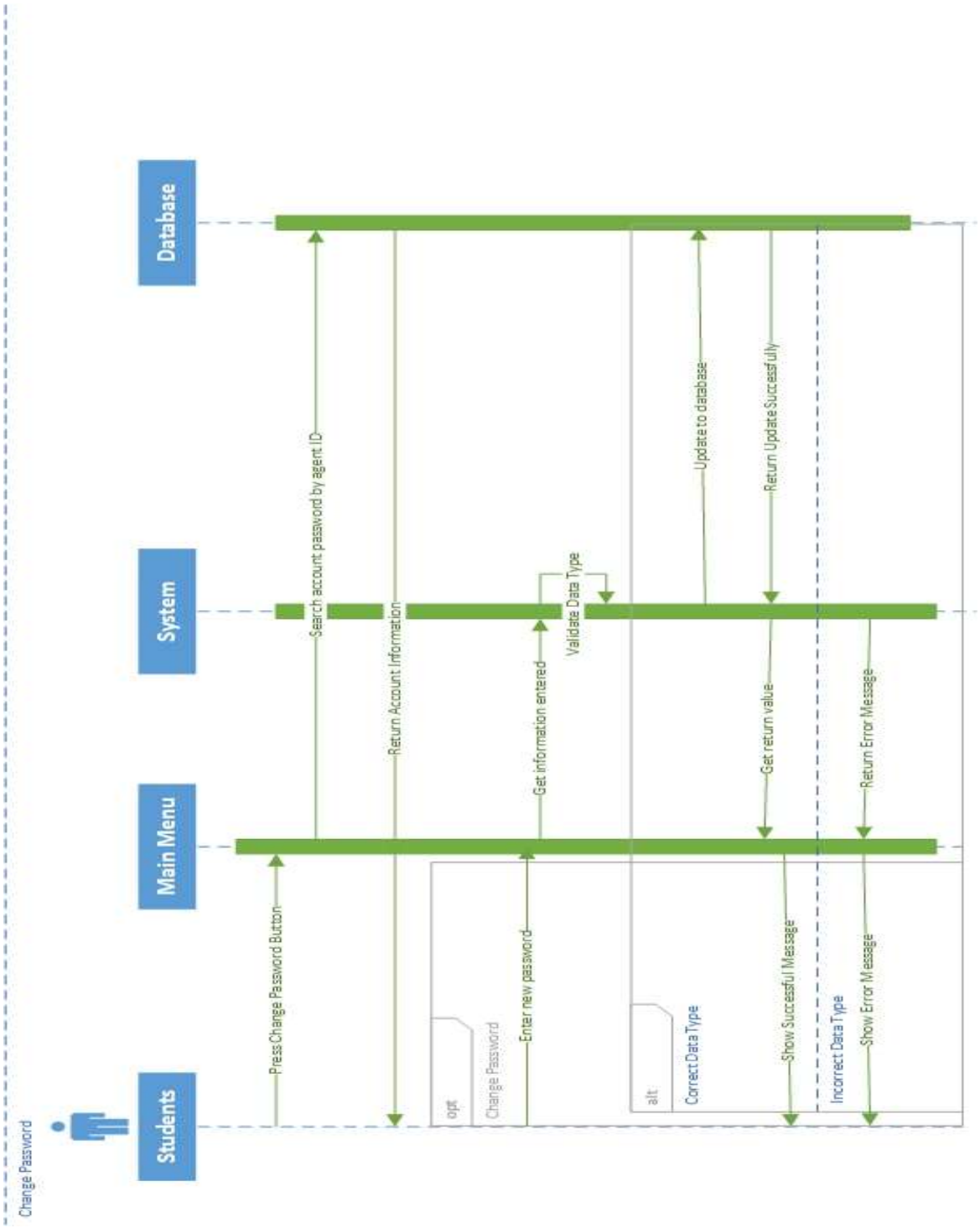
3.3.1 Logins



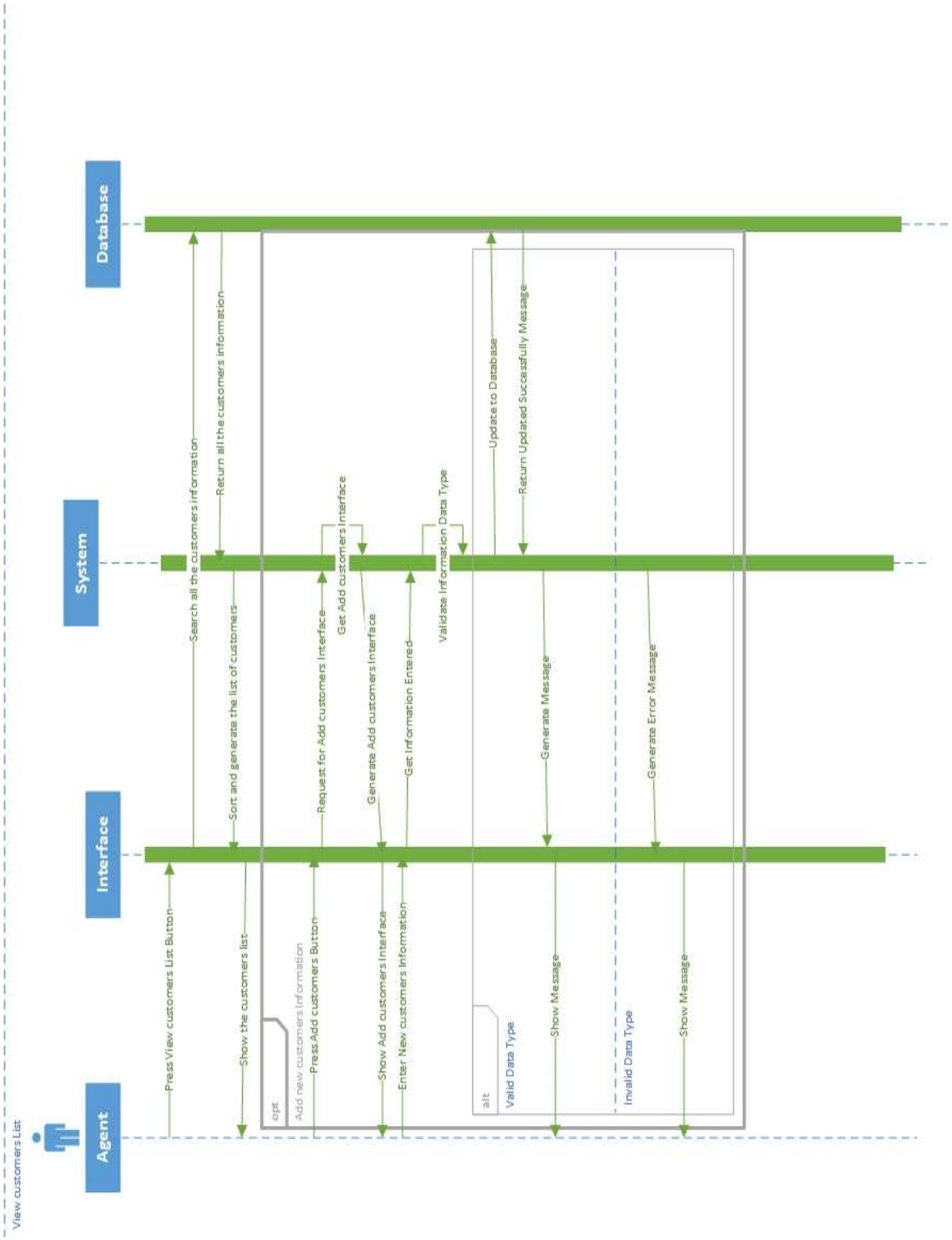
3.3.2 View Schedule(Agents)



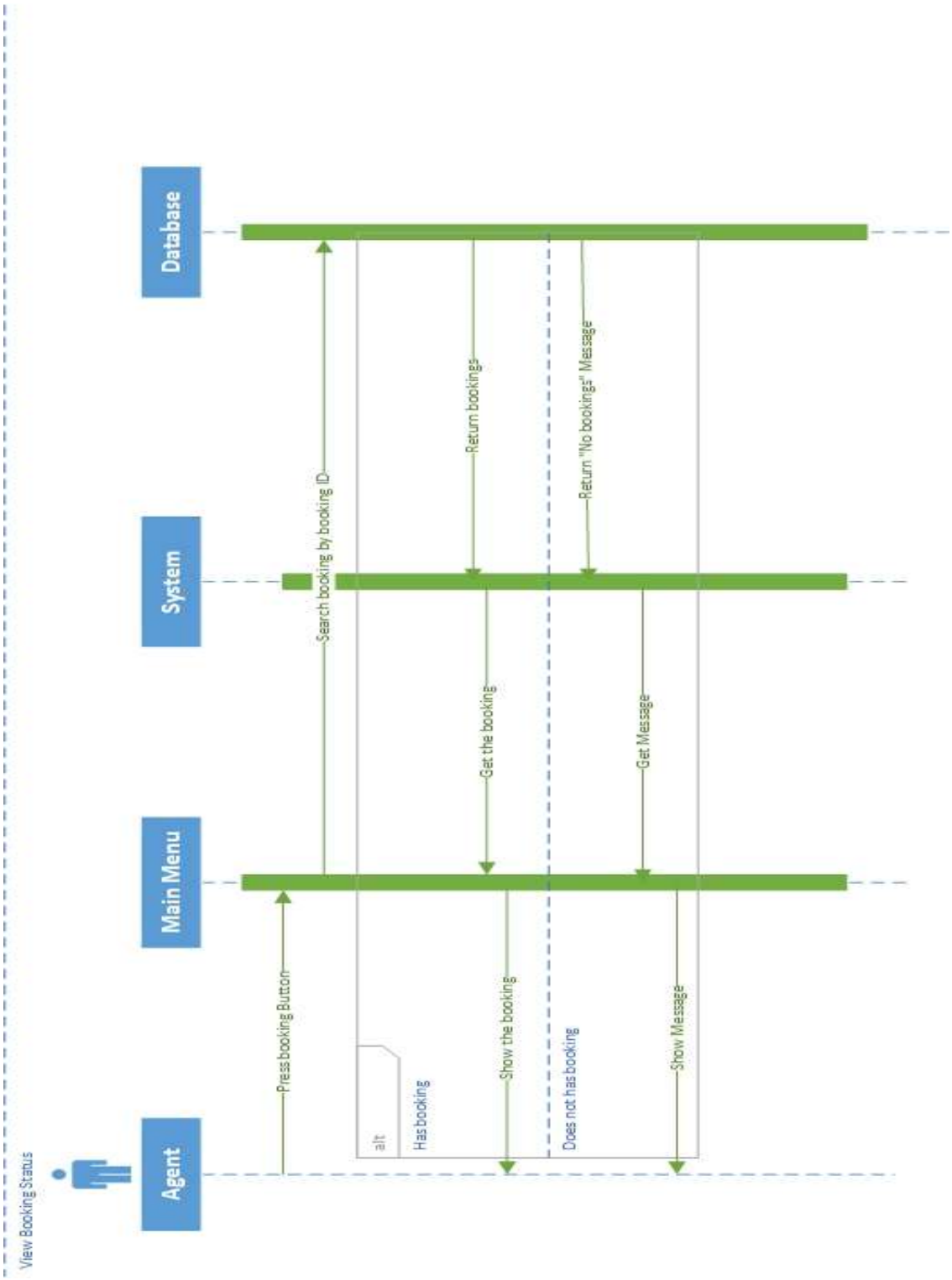
3.3.3 Change Password (Agents)



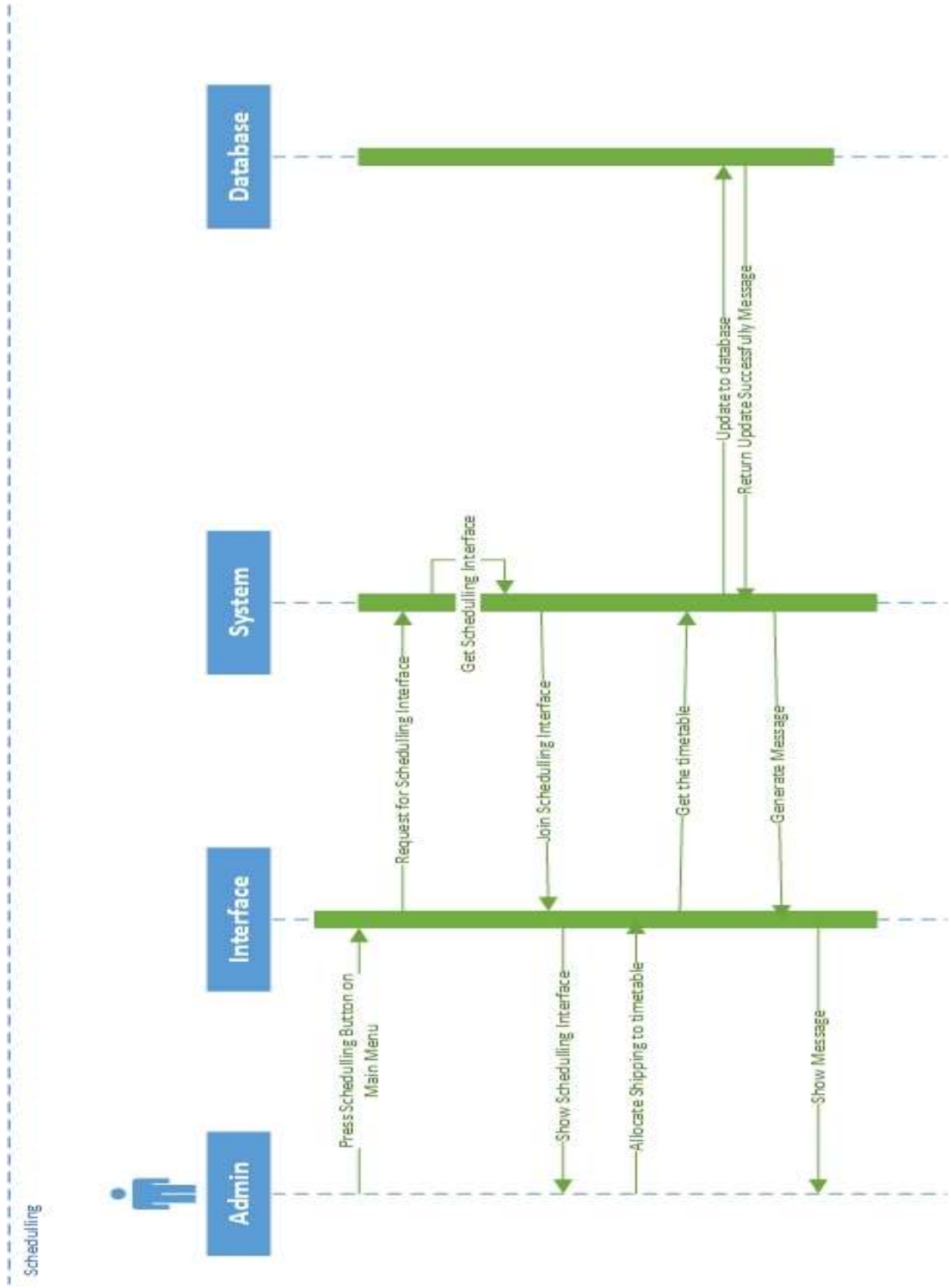
3.3.4 View Customers (Agents)



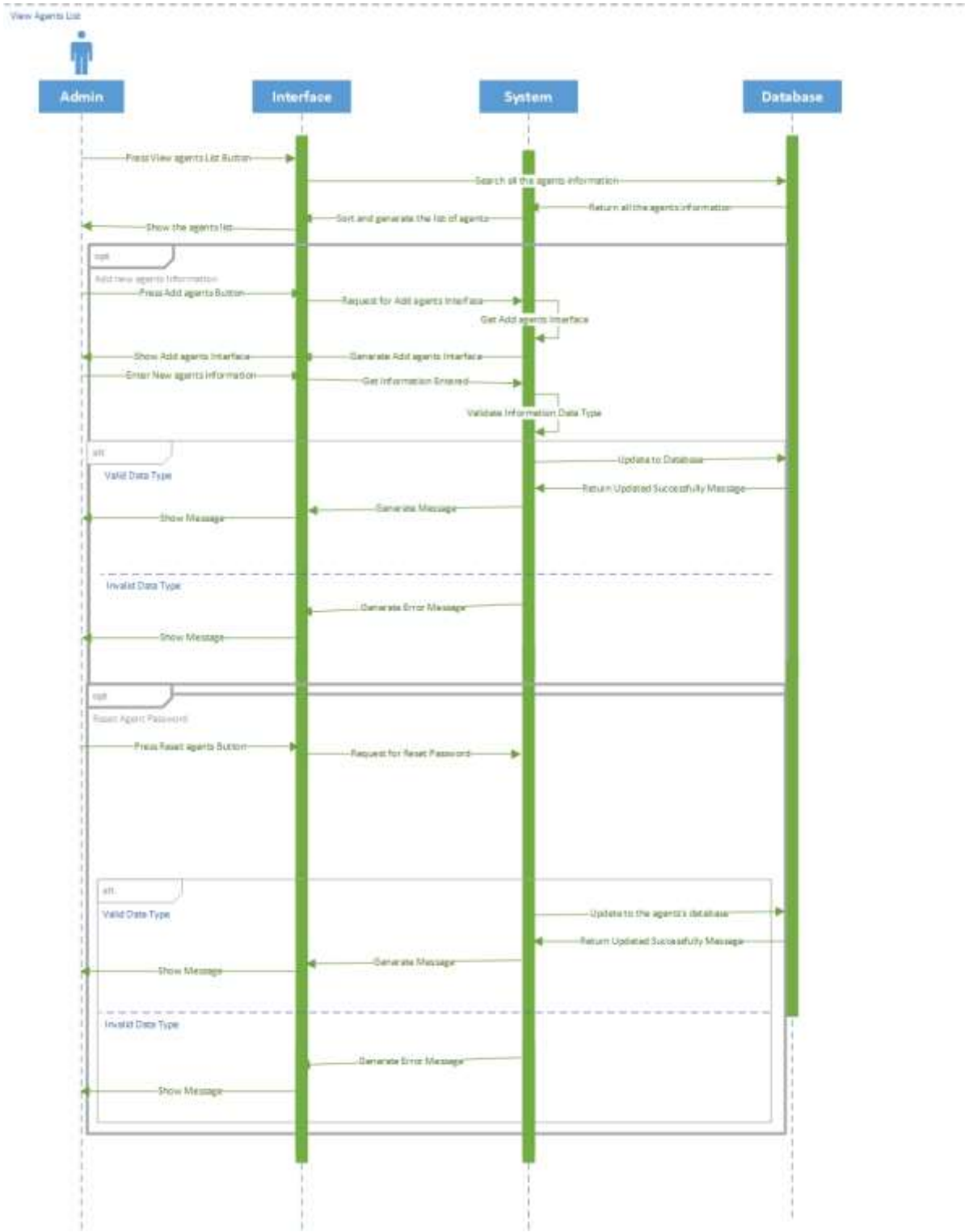
3.3.5 View Bookings (Agents)



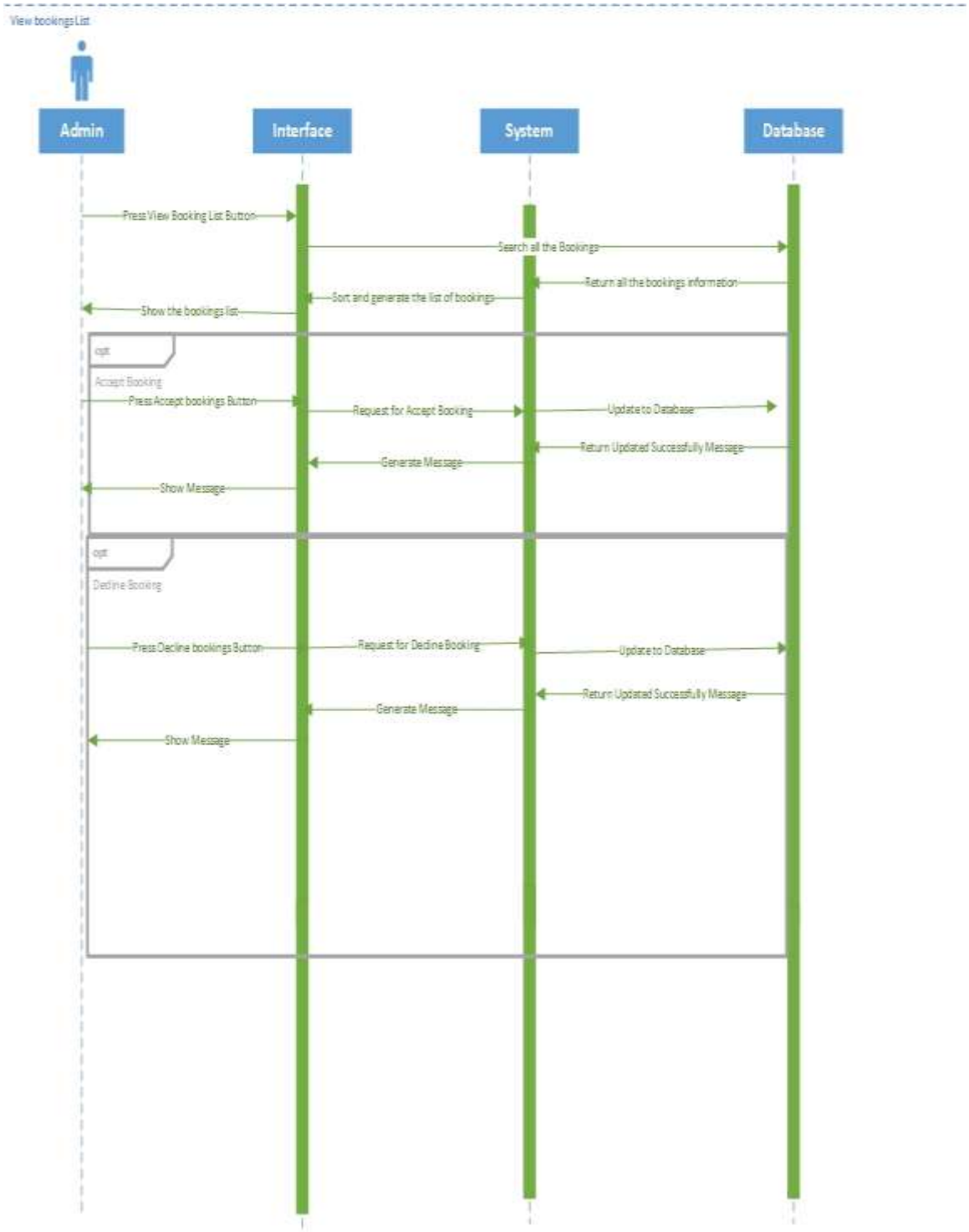
3.3.6 View Schedule(Admins)



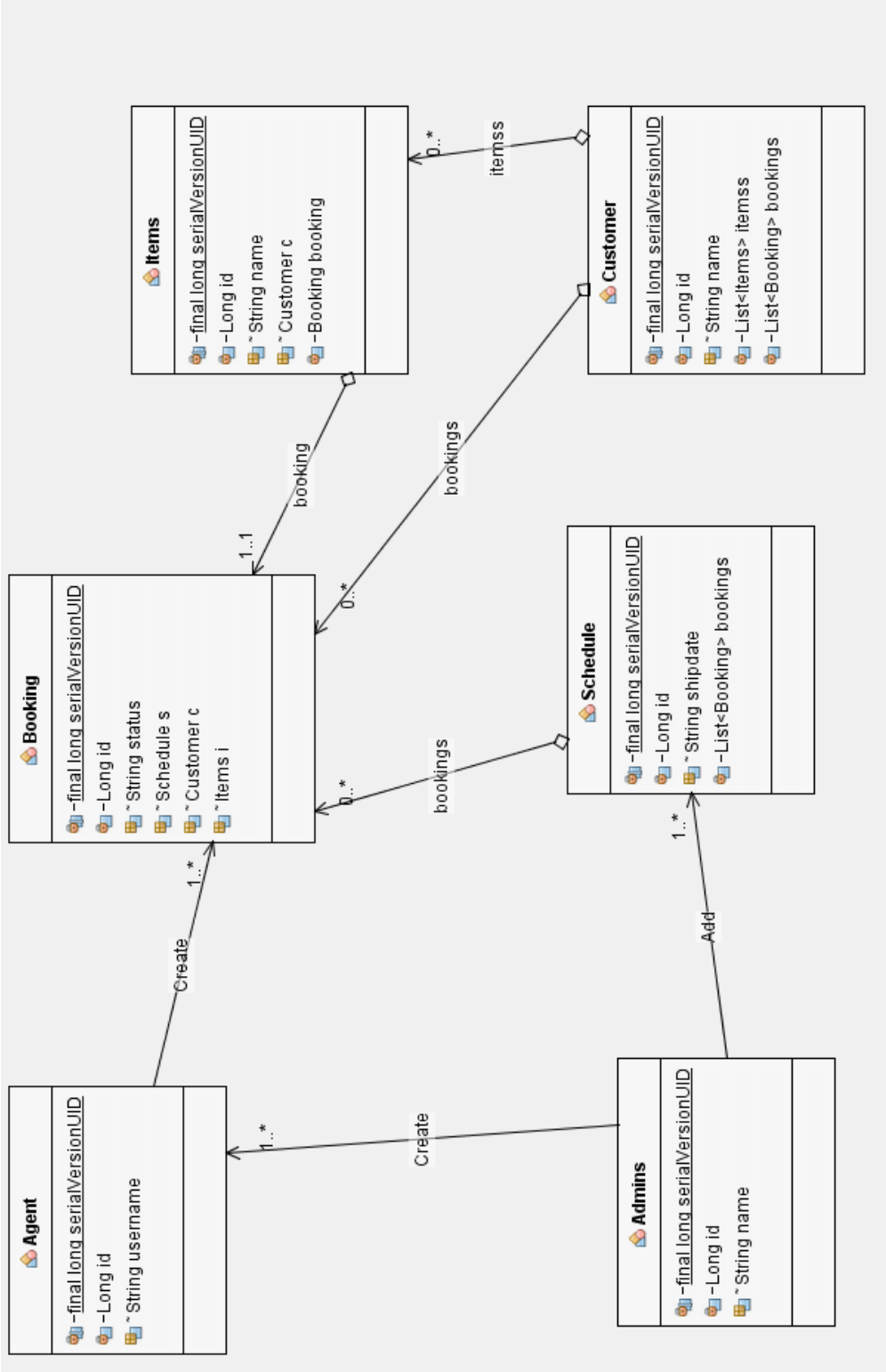
3.3.7 View Agents(Admins)



3.3.8 View Bookings (Admins)



3.4 Class Diagram



3.5 Data Dictionary

UserAccount				
Comment:	To store the login detail of Admin Account			
Field Name	Data Type	Field Length	Constraint	Description
ID	INT	*	Primary Key	User id, Auto Generated by system
NAME	VARCHAR	255	Unique	Username of user for login
PASSWORD	VARCHAR	255	Not Null	Password of user for login

Agent				
Comment:	To store the detail information of Agent Account			
Field Name	Data Type	Field Length	Constraint	Description
ID	INT	*	Primary Key	Agent id, Auto Generated by system
NAME	VARCHAR	30	Not Null	Agent Name
USERNAME	VARCHAR	20	Not Null	Agent Login UserName
NATION	VARCHAR	30	Not Null	Agent Nation
STATE	VARCHAR	30	Not Null	Agent State
PASSWORD	VARCHAR	30	Not Null	Agent Login Password

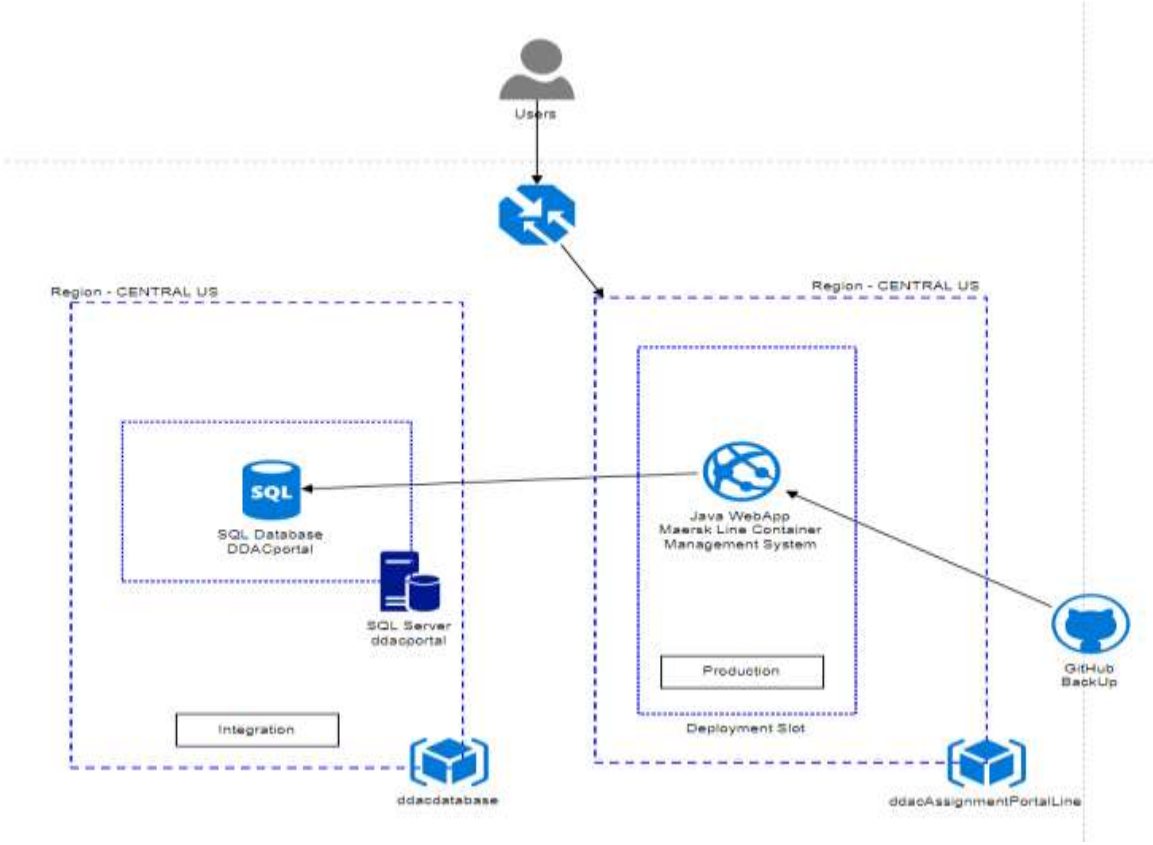
Schedule				
Comment:	To store the detail information of Shipping Schedule			
Field Name	Data Type	Field Length	Constraint	Description
ID	INT	*	Primary Key	Schedule id, Auto Generated by system
SOURCE	VARCHAR	30	Not Null	The place that shipping from
DESTINATION	VARCHAR	30	Not Null	The place that shipping to
SHIPDATE	VARCHAR	20	Not Null	The date of shipping
SHIPTIME	VARCHAR	20	Not Null	The actual time of shipping
CAPACITY	DOUBLE	*	Not Null	The capacity of shipping

Booking				
Comment:	To store the detail information of Bookings			
Field Name	Data Type	Field Length	Constraint	Description
ID	INT	*	Primary Key	Booking id, Auto Generated by system
STATUS	VARCHAR	10	Not Null	The status of the booking
CUSTOMER_ID	INT	*	Foreign Key	Customer ID, the customer own Booking
SCHEDULE_ID	INT	*	Foreign Key	Schedule ID, the schedule has Bookings
ITEM_ID	INT	*	Foreign Key	Item ID, Booking has Item.

Customer				
Comment:	To store the detail information of Customers			
Field Name	Data Type	Field Length	Constraint	Description
ID	INT	*	Primary Key	Customer id, Auto Generated by system
NAME	VARCHAR	40	Not Null	Customer Name
NATION	VARCHAR	30	Not Null	Customer Nation
ICNUMBER	VARCHAR	30	Not Null,Unique	Customer IC Number
GENTLE	VARCHAR	10	Not Null	Customer Gentle
PHONE	VARCHAR	*	Not Null	The Phone number of Customer
ADDRESS	VARCHAR	30	Not Null	The address of Customer

Items				
Comment:	To store the detail information of booking items			
Field Name	Data Type	Field Length	Constraint	Description
ID	INT	*	Primary Key	Item id, Auto Generated by system
ITEM_NAME	VARCHAR	40	Not Null	The name of Items
ITEM_CATEGORY	VARCHAR	40	Not Null	The category of Items
ITEM_QUANTITY	INT	*	Not Null	The quantity of item to be shipped
ITEM_WEIGHT	DOUBLE	*	Not Null	The total weight of items
CUSTOMER_ID	INT	*	Foreign Key	Customer ID, customers own the items

3.6 Cloud Architecture



The initial implementation of the Maersk Line Container Management System in the cloud environment focused primarily in the Central US region, and developer placed the main web application in one of the resource groups. The GITHUB will be holding the entire system as a backup for unexpected system crash. Among the services registered in the web application, there is a web service that provides a container for holding the Maersk Line Container Management System, and services can be scaled out according to the server's performance needs and future needs. Besides, the Azure database in a SQL Server database is another important function necessary to keep the data necessary for daily operation. Because of the first method of the database is applied during development, SQL Server is set to a different resource group, and the web application can only connect to the SQL database after setting up the firewall rule in the SQL Server to

achieve data security.

3.7 Design Consideration

3.7.1 Data information could be achieved the accurate

In the design of the website, the developer sets all information fields and performs text validation to prevent empty data types or incorrect data types from storing information. The Empty Textbox field will cause all users to have a common error in all applications. Therefore, the developer designs all web applications in the input text field using the verification method as a diagram design to avoid blank mistakes in the input text field.

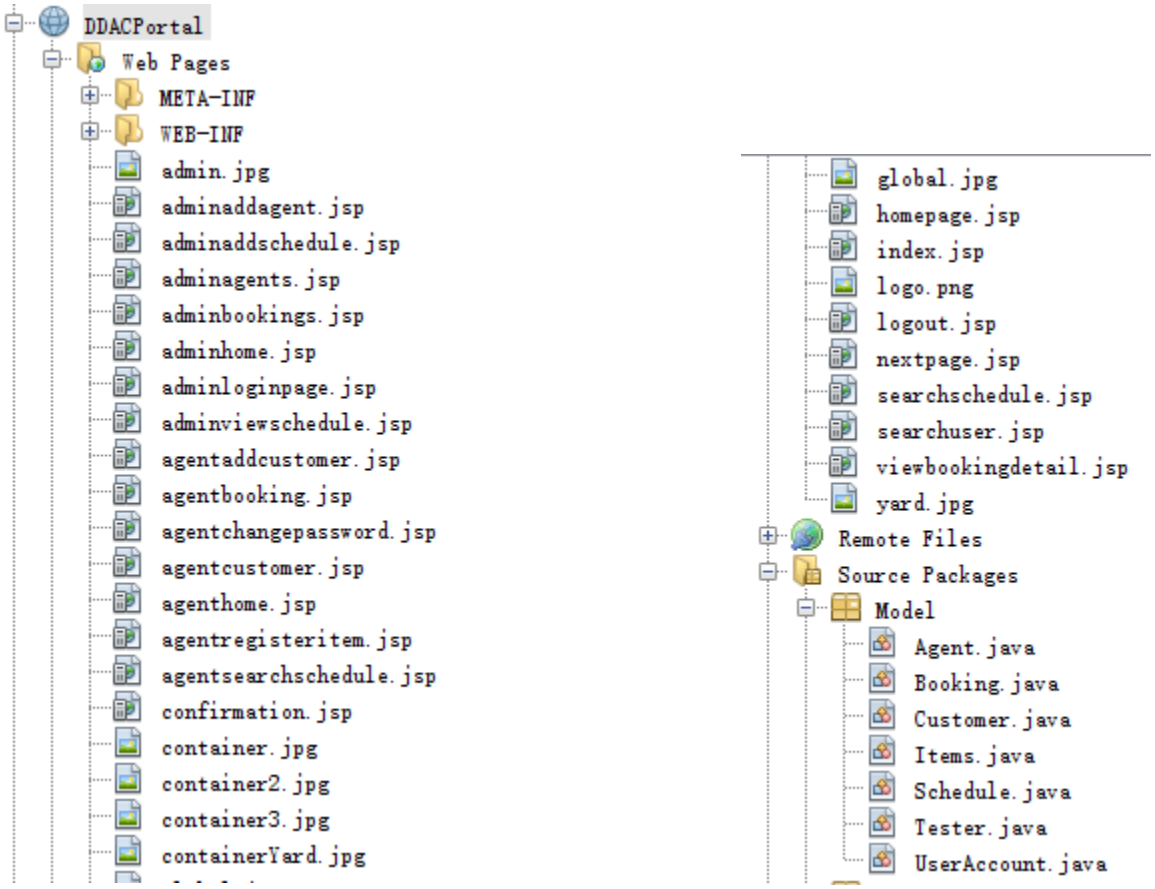
3.7.2 HTTP Sessions Not Persisted or Replicated

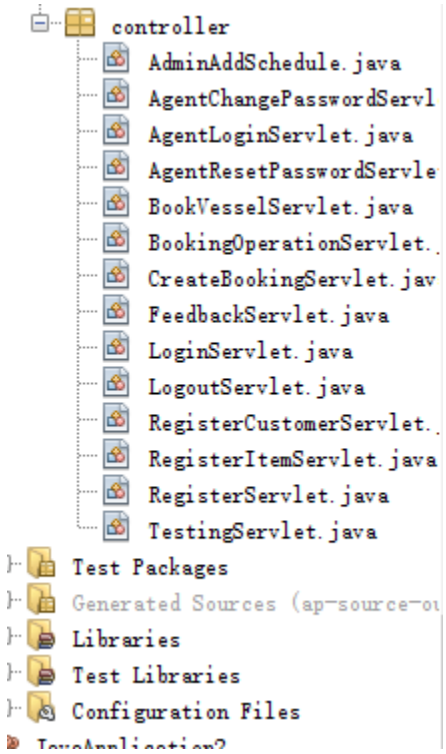
The Maersk Line Container Management System must support session affinity or sticky sessions for incoming HTTP requests to applications when session cookies are used. If multiple instances of an application are running on Maersk Line, all requests from a particular client are routed to the same application instance. This allows the web application container and framework to store session data specific to each user session. Maersk Line does not persist or replicate HTTP session data. When a user session persistent of an instance termination or crash makes another HTTP request, the request is routed to another instance of the application.

4.0 Implementation

4.1 Application Development

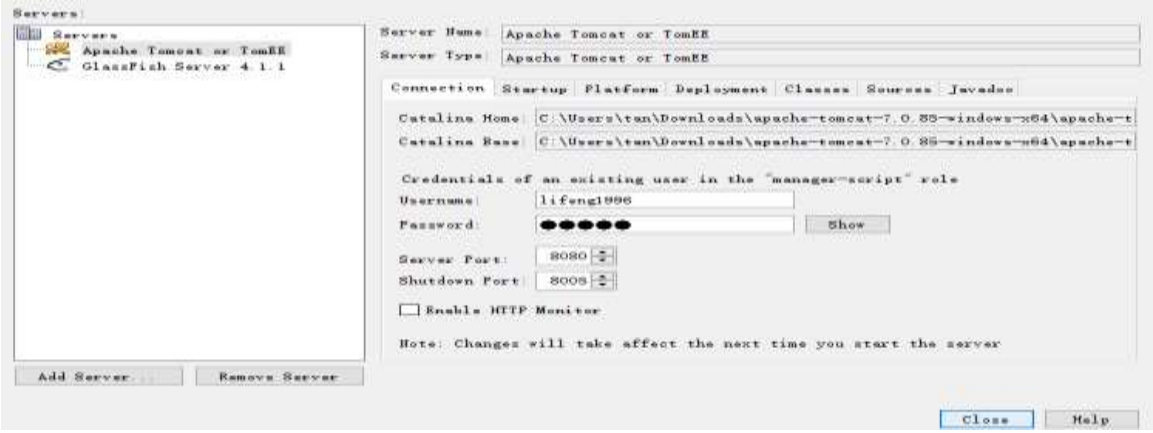
Development of the Maersk Line Container Management System is implemented in Java Programming Language and using the HTML to implement the user interfaces. In Java Web app development environment, it mainly breaks down the application into three files which are Model, Controller and WebSites. Besides, this web application needs to integrate with SQL database to store and retrieve the data needed.





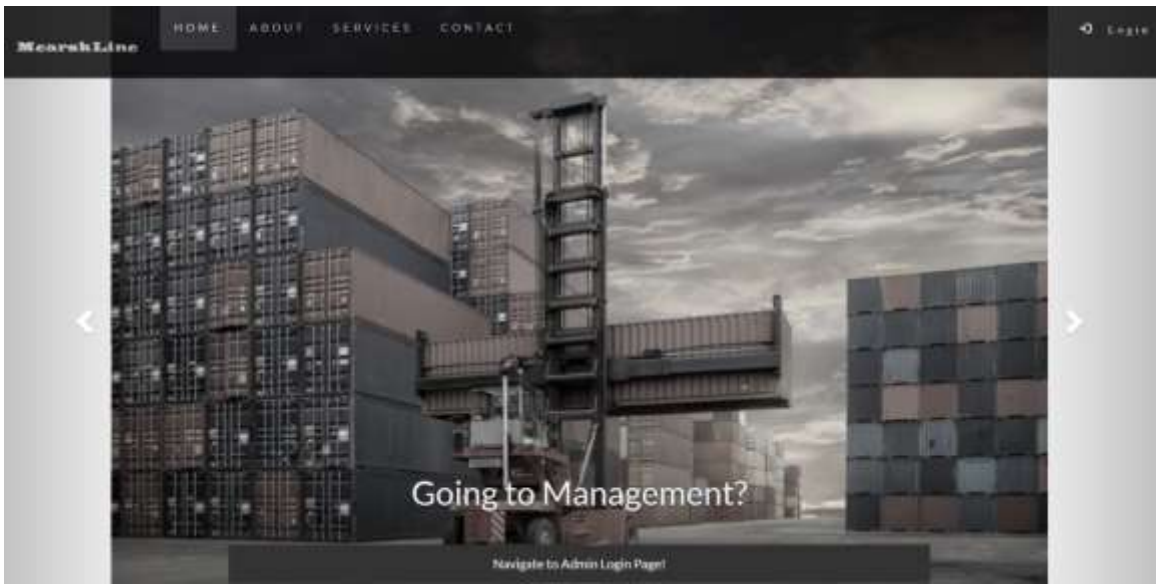
The Model file is used to store the Entity Classes that created for the system; The controller file is storing the Java Servlet which is used to process some of the backend progress in the system; While the WebSites file is holding all the user interfaces file (jsp files).

Besides, most of the design and function of the websites are using the Bootstrap and Javascript. As the prerequisite of the web application development, the server container is playing an important rule which enable a web application can be run well. The web server container is decided to user Apache Tomcat 7.0 since this version is relatively more stable than the other latest version.



4.1.1 User Interface Example (Bootstrap)

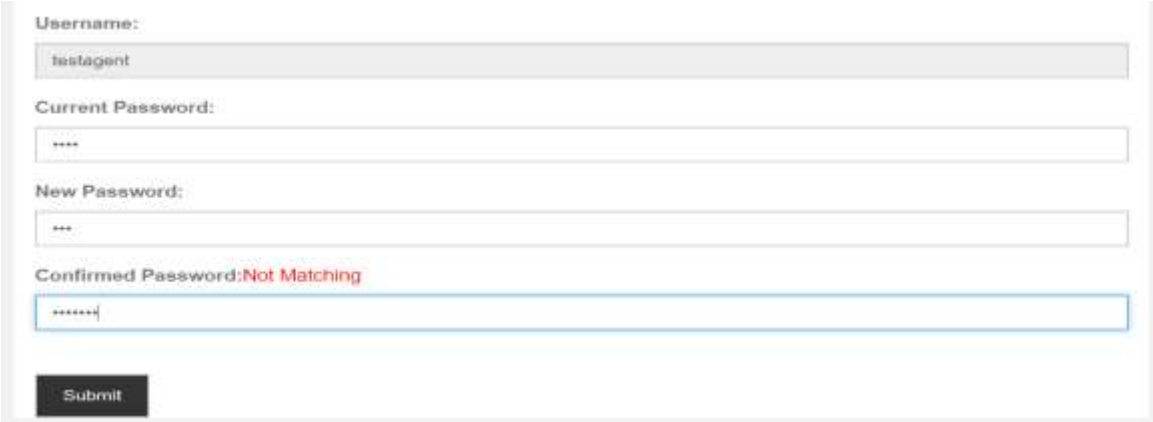
Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. It allows developers can quickly prototype ideas and build the whole application quickly using Sass variables and mixins, highly responsive grid systems, rich embedded components, and powerful plugins built with jQuery (Bootstrap, 2018). In the web application development, Bootstrap provides a convenience for developers to structure the web site layout.



```
<title>Maersk Line-Agent-Welcome</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet" type="text/css">
<link href="https://fonts.googleapis.com/css?family=Montserrat" rel="stylesheet" type="text/css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

4.1.2 Example of Password Validation Function (JavaScript)

JavaScript (JS) is a lightweight, interpreted programming language with first class features (Mozilla, 2018). In the browser environment best known as the scripting language of the web page, it uses this environment other than many browsers, such as node.js and Apache CouchDB. JS is a prototype-based multi-paradigm dynamic scripting language that supports object-oriented, imperative, declarative style (such as functional programming) style (Mozilla, 2018). In this web application development, JavaScript is being used frequently to achieve several real-time and dynamic functions without communicate to the Java Servlet such as password checking, table filtering and Username Availability.




```
function validatePassword() {
var pass2 = document.getElementById("loginpwd").value;
var pass1 = document.getElementById("conpwd").value;
if (pass1 != pass2)
        document.getElementById("conpwd").setCustomValidity("Passwords Don't Match");
else
        document.getElementById("conpwd").setCustomValidity('');
//empty string means no validation error
}
function checkoripassword()...9 lines
function checkbeforesubmit()...5 lines
function check() {
if (document.getElementById("loginpwd").value ===
        document.getElementById("conpwd").value) {
document.getElementById("message").style.color = 'green';
document.getElementById("message").innerHTML = 'Matching';
document.getElementById("submitbutton");
} else {
document.getElementById("message").style.color = 'red';
document.getElementById("message").innerHTML = 'Not Matching';
}
}
```

4.1.3 Example Code of Register Agent Account to Database (Servlet)

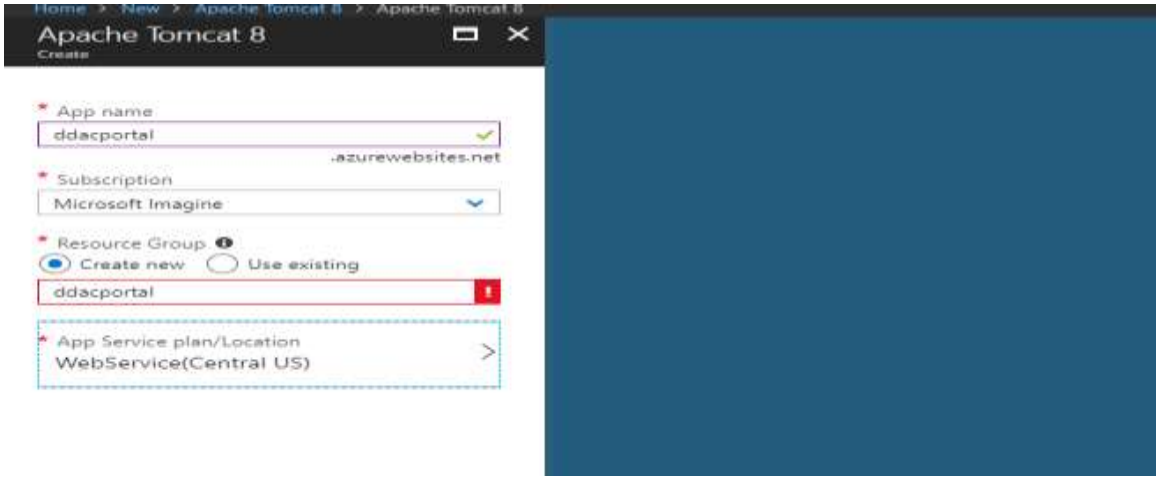
The servlet code is designed to get all the information inserted by the user from the website and process the registration. Firstly, the system will get the username and check the availability of the username in database. If the username is available, the system will create the account and update to the database; Else the system will show “Username is not available” message to the user.

```
        String name = request.getParameter("reginame");
        String nation = request.getParameter("nation");
        String state = request.getParameter("state");
        PrintWriter out = response.getWriter();
        try {
          Connection conn = null;
          Statement stat = null;
          Agent ag = new Agent(email, name, password, nation, state);
String driver ="com.microsoft.sqlserver.jdbc.SQLServerDriver";
        try {
            try {
          Class.forName(driver);
        } catch (ClassNotFoundException e) {
          e.printStackTrace();
        }
          String dbURL = "jdbc:sqlserver://ddbcportal.database.windows.net:1433;databaseName=DDBCportal";
          String user = "lifeng1996";
          String pass = "Ag@jfp4!";
          conn = DriverManager.getConnection(dbURL, user, pass);
          System.out.println("here");
          out.println("here");
          stat = conn.createStatement();
          String sql;
          sql = "INSERT INTO Agent(USERNAME,NATION,STATE,NAME,PASS) VALUES ('"+ag.getUsername()+"','"+ag.getNation()+"','"+ag.getState()+"','"+ag.getName()+"','"+ag.getPass()+"')";
          stat.executeUpdate(sql);
          stat.close();
          out.println("<script type=\"text/javascript\">");
  out.println("alert('Account is created successfully');");
  out.println("location='adminagents.jsp' ");
  out.println("</script>");
```

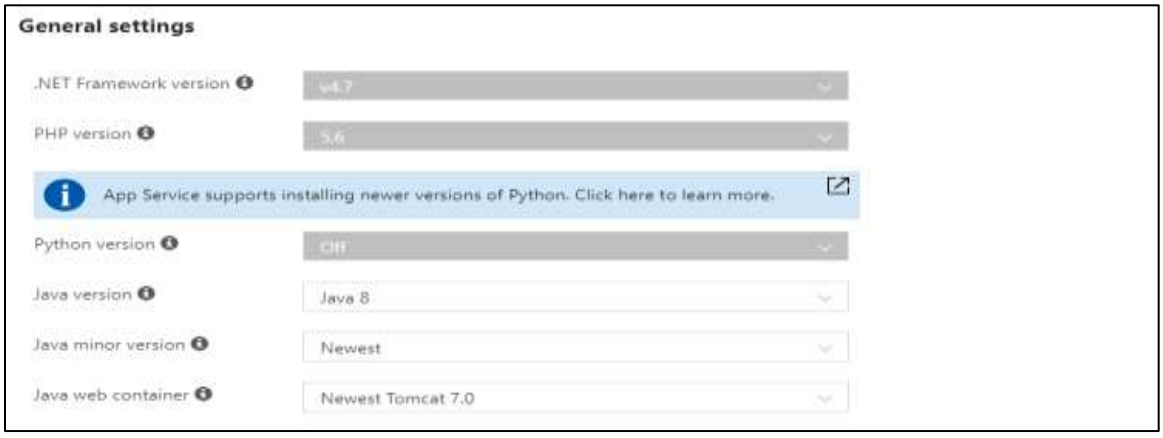
4.2 Azure Publishing

4.2.1 Create New Web Service On Azure

First, create a new Tomcat Web Server Container, register the application container, obtain the application name, subscription, resource group, application service plan and region, operating system needed. The important point is that because the application name is unique and it is necessary to analyze the visitor flows in different areas appropriately, in order to maintain the reliability of the application, the pricing layer most suitable for the Web application is registered.



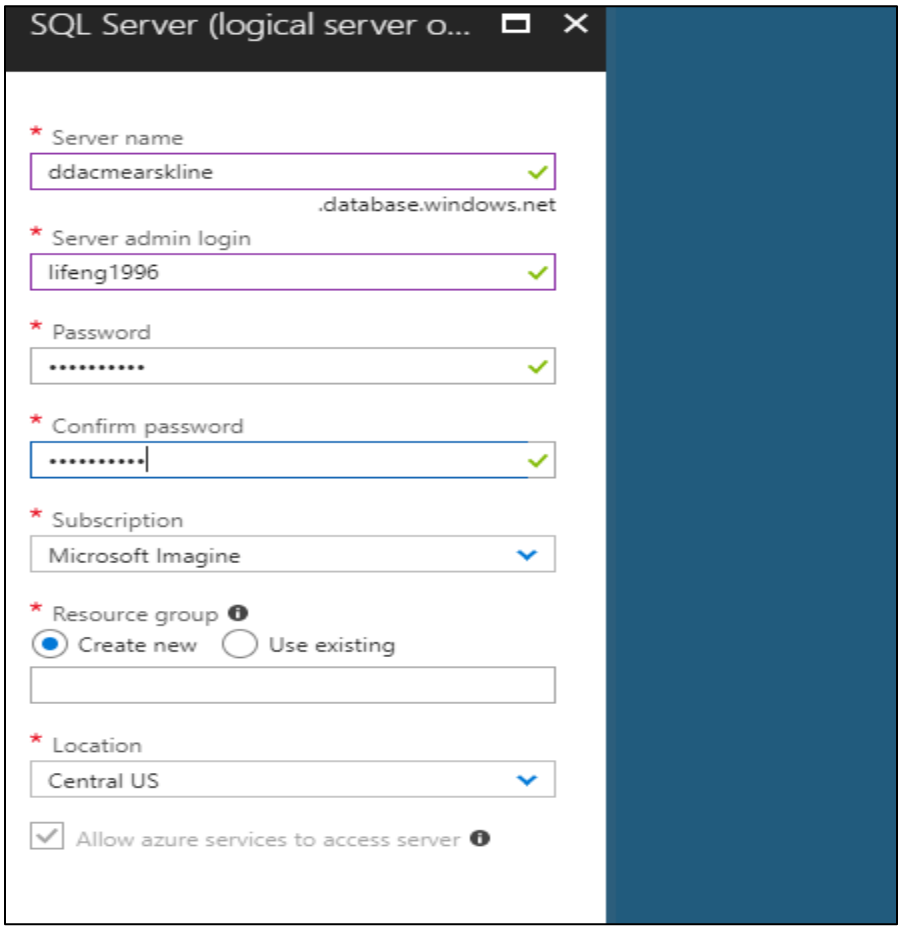
Besides, since this project system is developed in Java Language hence a proper setup of Java Language Environment is required and important in the web service configuration.



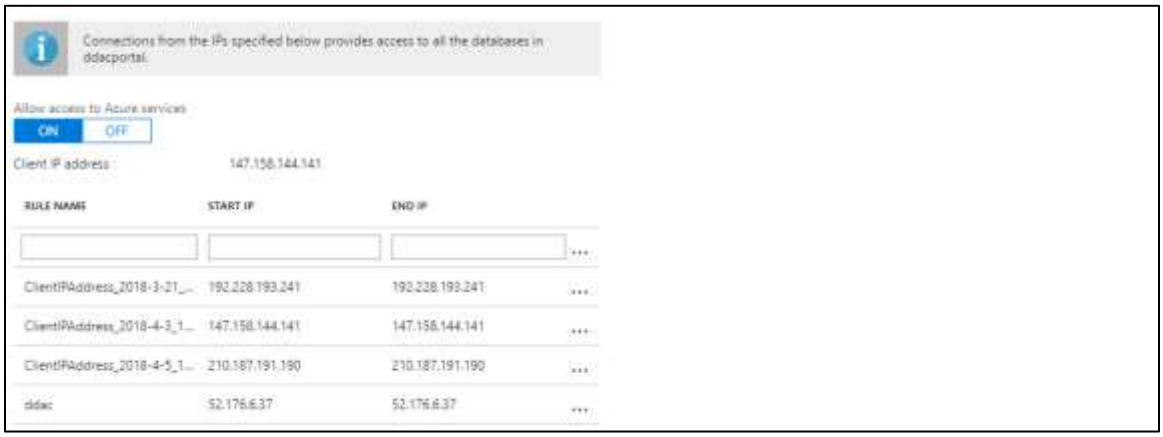
4.2.2 Azure SQL Server and Database Setup

Next, set up an SQL database to store and retrieve data for web applications. The server is also created here and hosts databases of a specific region with the user name and password assigned to access the server.

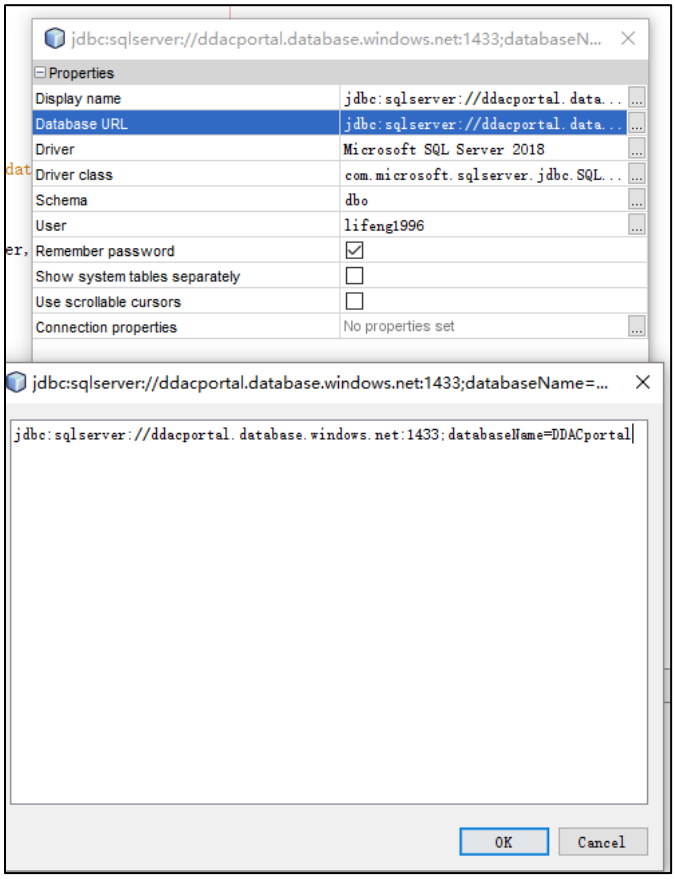
This requires information on database name, subscription, resource group, source, server, and price tier. The database name must be unique among the individual Azure databases.



Once setup the database server, to allow the web application to connect the server, firewall setting to allow the web application server connects the database server is required.



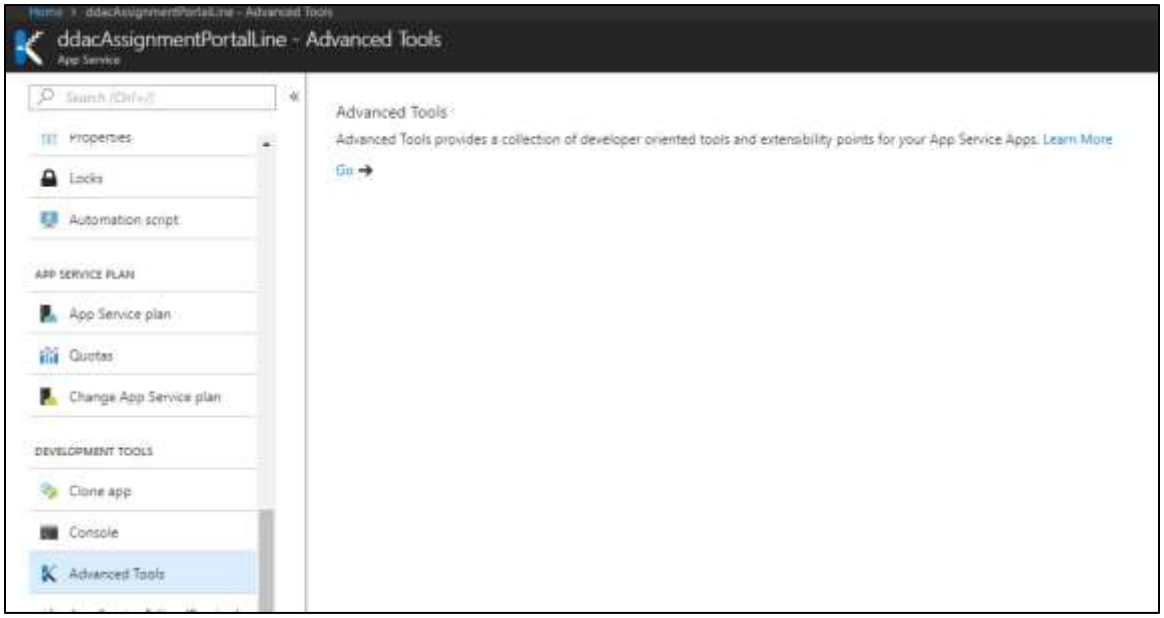
In the management page of SQL database, copy the connect string which allows web application to connect the target database correctly.



4.2.3 Web Application Publishing

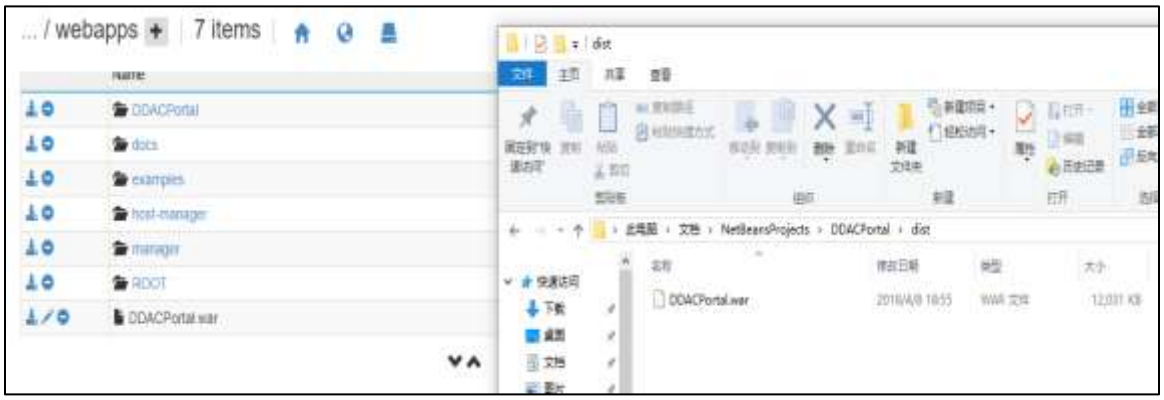
The last step is to publish the completed web application to the prepared container in the Azure. In Java Web Application, there is a WAR file will be built and stored in “dist” file. This WAR file is a compressed web application file.

To deploy the web application by WAR, it can be done easily by entering the “Kudu Service” which can be accessed from the “Deployment Tool” of the App Service



Then, go to the “Debug Console” and choose either CMD or PowerShell. Enter the command “cd home\site\wwwroot\bin\apache-tomcat-8.5.24\webapps”. Then just “Pull

and Drop" the WAR file in the file.

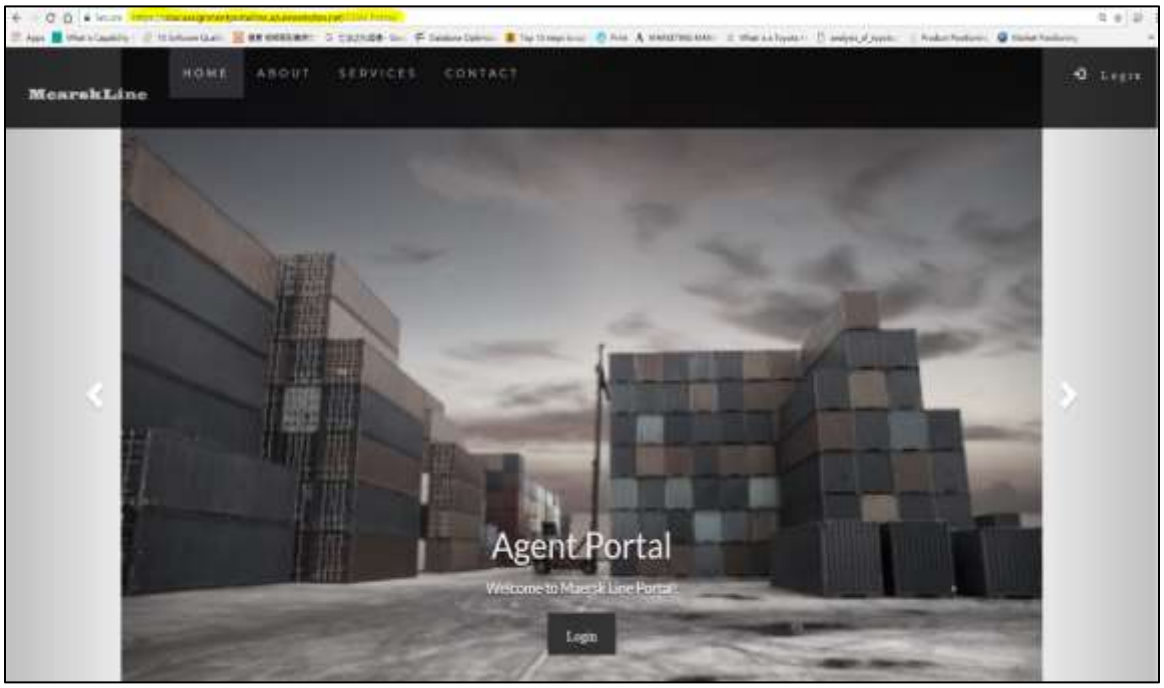


After uploaded successfully, it will be unzipped automatically and now is able to browse the web application.

GitHub Link: https://github.com/lifeng1996/DDACportal

Azure Link: https://ddacassignmentportalline.azurewebsites.net/DDACPortal/

Microsoft Stream Link: https://web.microsoftstream.com/video/216c6811-0a1e-4d29-8b28-4fe88970af74



4.3 Application Scaling

Application scaling allows the application to respond to changes in traffic and automatically allocate the resources needed to process the current request (Cavale, 2017). Azure Web App can monitor incoming web traffic and automatically add or delete application gears to accommodate changes in demand.

4.3.1 Web App Scale

Among the pricing plans listed above, there are basic categories and standard ones that provide different offerings. The pricing plan of the web app service is decided to apply for the Standard S1 with the following reasons. First, compare with the other standard plans, there are nothing different between three of them but only the CPU and RAM. Moreover, since the system will be tested for a given period hence it is not a large scale of system in early stage. The standard plan provides a medium storage which has capability than a 50 GB and able to handle large volumes of transactions within a certain period of time. Besides, by providing custom domains and SSL in both plans, the

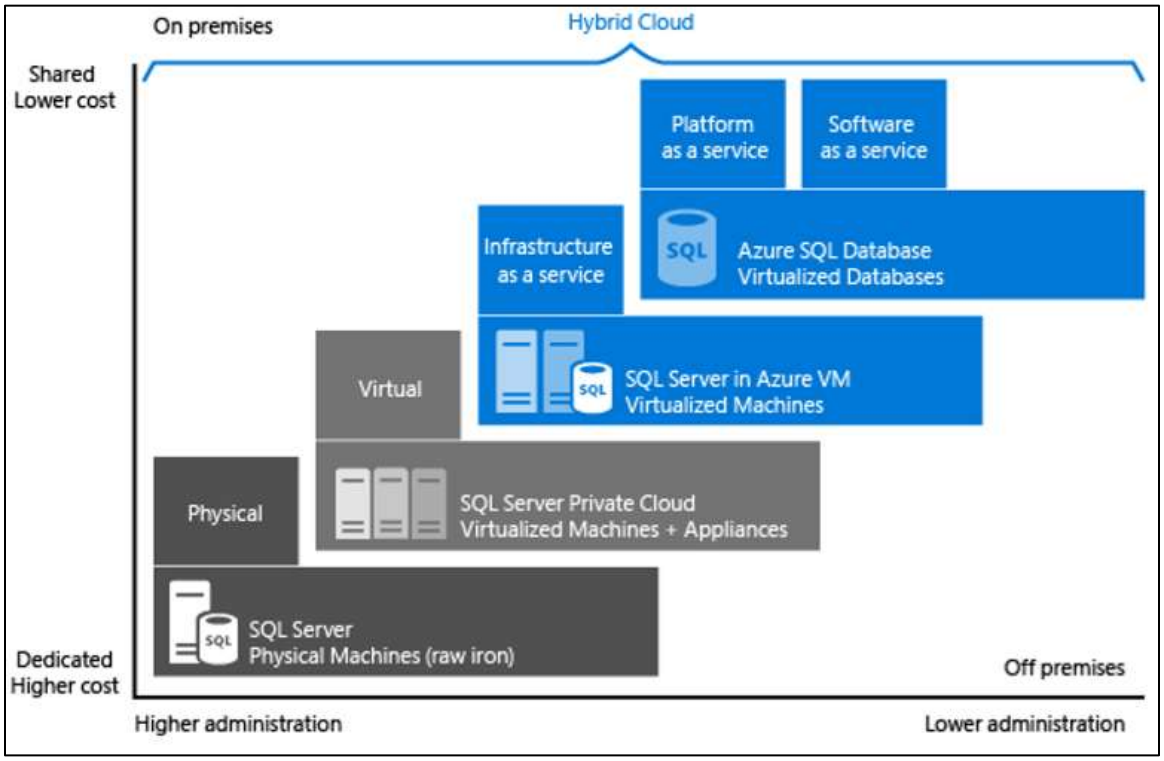
standard plan provides special advantages and SSL support. In addition, autoscale is also a general offer, but up to 10 instances can be extended with the standard plan. The other additional feature provided only in the standard plan that backs up application settings and data daily to ensure data consistency. Furthermore, in preparation for unexpected situations, the extra five deployment slots can be used to hold the recent deployment restore points. In addition, traffic managers can handle stable connections between clients and servers by connecting to the closest server in the region.

4.3.2 SQL Database Scale

In this deployment, Azure SQL Server also uses standard plans as well. Since the system will be tested for a given period and it is not requiring high computation and storage for the testing data hence a standard plan with 100GB of storage and 10 compute units are enough for the current system. Besides, the storage and compute units of the database can be scaled up in future when the system is getting mature and requiring a higher performance of database.

4.4 Managed Database (Paas)

Due to the widespread use of online cloud and management services, services previously available only on locally provisioned servers became available by clicking buttons. The concept of a platform as a service (PaaS) is a valuable tool for markets that are trying to thoroughly cut development and deployment time.



According to Microsoft (Microsoft, 2018), Azure SQL database enables development of platform (DBaaS), SaaS (as-a-service) application as service (PaaS) database or database (DBaaS). It provides compatibility with most SQL Server features. There is nothing different from the general SQL Server available on physical premises machine, private cloud environment, third party private cloud environment, and public cloud (Microsoft, 2018). In these environments, the same server products, development tools, and sets of expertise are provided.

There are several advantages to using PaaS instead of local server. First, the speed of

provisioning increases the agility of innovation (Yard, 2018). With PaaS, developers can create 10 servers using one script and delete servers after 5 minutes testing (Cavale, 2017). Also, developers can reduce operating costs. Developers can reduce costs by using features such as autoscaling (Cavale, 2017) with the Microsoft Azure service and provisioning only the resources organization needs.

An Azure managed database is suitable for new cloud-designed applications that have time constraints in development and marketing (Microsoft, 2018). Besides, teams that need built-in high availability, disaster recovery, and upgrade for the database while also do not want to manage the underlying operating system and configuration settings are recommended to utilize the Azure Managed Database (Microsoft, 2018).

Moreover, since the Azure Managed Database provides a well-structured and highly learnability resources (Microsoft, 2018). It is mainly focus on the application layer while does not want to employ IT resources for configuration and management of the underlying infrastructure (Microsoft, 2018). By eliminating the hardware costs and administrative costs, a cost saving benefit can be achieved by applying Azure Managed Database (Microsoft, 2018).

For business, it is providing the built-in fault tolerance infrastructure capabilities, automated backups, point in time restore, geo restore and active geo replication to improve business performance (Microsoft, 2018). These characteristics enable a business to be more sustainable (Microsoft, 2018). Meanwhile, the high efficiency and convenience of the Azure Managed Database makes On-premises application can access data in Azure SQL Database wherever and whenever (Microsoft, 2018).

5.0 Testing

5.1 Unit Testing

Unit Testing will be performed to test each of the functions of the system. It is carried out by entering the input and check for the validation of the output of system. By gathering the result of the testing, developer will able to figure out the shortage of the system. In unit testing, each of the functions of the system will be tested separately to see whether it shows the exactly output as the deliverables.

Test ID: 1

Test Description: Login to the system with registered account

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T1.1	Login to the system with valid	1. Navigate to 2. Enter Input 3. Submit the	1. Username: test 2. Password: test	1. Show the right home page according	1. Show the right home page according	Pass
T1.2	Login to the system with invalid account username.	1. Navigate to Login Modal 2. Enter Input 3. Submit the	1. Username: test123 2. Password: test	1. Show "Wrong username or password" message and reload the	1. Show "Wrong username or password" message and reload the homepage.jsp	Pass
T1.3	Login to the system with invalid account password	1. Navigate to Login Modal 2. Enter Input 3. Submit the	1. Username: test 2. Password: test123	1. Show "Wrong username or password" message and reload the	1. Show "Wrong username or password" message and reload the homepage.jsp	Pass

Admin

Test ID: 2

Test Description: View Schedule

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T2.1	View Schedule	1. Press "Shipping Schedule" link which at the top		1. Redirect to the adminschdule.jsp	1. Redirect to the adminschedule.jsp	Pass

Test ID: 3

Test Description: Add Schedule

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T3.1	Add New Schedule by inserting correct data type to all editable field	1. Press "Add Schedule" button which at the schedule page	1. Source: Malaysia	1. Update new information to database	1. Update new information to database	Pass
		2. Enter Input	2. Destination: Taiwan	2. Show "Created Successfully" and redirect to adminhome.jsp	2. Show "Created Successfully" and redirect to adminhome.jsp	Pass
		3. Submit the	3. Shipping Date: 11/4/2018			
			4. Shipping Time:1830			
			5. Capacity: 5000kg			
T3.2	Add New Schedule by inserting data to few editable field only	1. Press "Add Schedule" button which at the schedule page	1. Source: Malaysia	1. Show "Data has not inserted completely" message	1. Show "Data has not inserted completely" message	Pass
		2. Enter Input	2. Destination: Taiwan			
		3. Submit the	3. Shipping Date: 11/4/2018			
			4. Shipping Time:1830			
			5. Capacity: null			
T3.3	Add New Schedule by inserting wrong data type	1. Press "Add Schedule" button which at the schedule page	1. Source: Malaysia	1. Show "Wrong data type inserted"	1. Show "Wrong data type inserted"	Pass
		2. Enter Input	2. Destination: Taiwan			
		3. Submit the	3. Shipping Date: 11/4/2018abc			
			4. Shipping Time:1830			
			5. Capacity: 5000kg			

Test ID: 4

Test Description: Reset Password

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T4.1	Reset Agent Account Password	1. Press "Reset Password" On the target Agent Account		1. Update new password to database	1. Update new password to database	Pass
				2. Show "Updated Successfully" and redirect to adminhome.jsp	2. Show "Updated Successfully" and redirect to adminhome.jsp	Pass

Test ID: 5

Test Description: View Agents

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T5.1	View Registered Agents	1. Press "Agents" link which at the top nav-bar		1. Redirect to the adminagent.jsp	1. Redirect to the adminagent.jsp	Pass

Test ID: 6

Test Description: Add New Agents

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T6.1	Add Agents by inserting correct information in all editable field	1. Press "Add New Agent" button which in page. 2. Enter Input 3. Submit tha	1. Username: testagent 2. Agent Name: Tan Li Feng 3. Nation: Malaysia 4. State: Johor 5. Password: test 6. Confirmed Password: test	1. Agent Account is created successfully in database. 2. Show successful message and redirect user to adminhome.jsp	1. Agent Account is created successfully in database. 2. Show successful message and redirect user to adminhomepage.jsp	Pass Pass
T6.2	Add Agents by inserting wrong data type and leave some blanks in some editable field.	1. Press "Add New Agent" button which in page. 2. Enter Input 3. Submit tha	1. Username: testagent 2. Agent Name: Tan Li Feng 3. Nation: Malaysia 4. State: Johor123 5. Password: test 6. Confirmed Password: null	1. Show "Please enter proper data type in state box" message 2. Show "Please make sure all information are inserted" message	1. Show "Please enter proper data type in state box" message 2. Show "Please make sure all information are inserted" message	Pass Pass

Test ID: 7

Test Description: View Bookings

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T7.1	View Bookings	1. Press "Bookings" button which at the top		1. Show all the bookings in list.	1. Show all the bookings in list.	Pass

Test ID: 8

Test Description: View Booking Detail

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T8.1	View Booking Detail	1. Press"View Detail" button which at the		1. Run viewbooking.jsp and show all details in new tab.	1. Run viewbooking.jsp and show all details in new tab.	Pass

Test ID: 9

Test Description: Accept bookings

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T9.1	Accept bookings	1. Press"Accept" button at the operation columns of the booking list		1. Updates the booking's status to database	1. Updates the booking's status to database	Pass Pass

Test ID: 10

Test Description: Decline bookings

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T10.1	Decline bookings	1. Press"Decline" button at the operation columns of the booking list		1. Updates the booking's status to database	1. Updates the booking's status to database	Pass

Employees

Test ID: 11

Test Description: Change Password

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T11.1	Change Password by inserting correct current password and correct confirmed password	1. Press "Change Password" to be navigated to Change Password 2. Enter Input 3. Submit the form	1. Current Password: test 2. New Password: test123 3. Confirmed Password: test123	1. Update new password to database 2. Show "Updated Successfully" and redirect to agenthome.jsp	1. Update new password to database 2. Show "Updated Successfully" and redirect to agenthome.jsp	Pass Pass
T11.2	Change Password by inserting wrong current password and different password in confirmed password field	1. Press "Change Password" to be navigated to Change Password 2. Enter Input 3. Submit the	1. Current Password: test12356 2. New Password: test123 3. Confirmed Password: test	1. Show "Wrong current password" 2. Show "Confirmed Password is not match with new password."	1. Show "Wrong current password" 2. Show "Confirmed Password is not match with new password."	Pass Pass

Test ID: 12

Test Description: Search Bookings

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Expected Result	Status(Pass/Fail)
T12.1	Search bookings by inserting keyword.	1. Navigate to adminhome.jsp 2. Enter keyword in the search input 3. Press "Search" button	1. Source: Taiwan 2. Destination: Malaysia	1. Run searchbooking.jsp and show the result in the content box.	1. Run searchbooking.jsp and show the result in the content box.	Pass
T12.2	Search bookings without inserting keyword.	1. Navigate to adminhome.jsp 2. Enter keyword in the search input 3. Press "Search" button	1. Source: Taiwan 2. Destination: NULL	1. Show "Please Insert completely criteria" message	1. Show "Please Insert completely criteria" message	Pass

Test ID: 13

Test Description: Place Booking

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T13.1	Apply booking	1. Select Schedule 2. Select Customers 3. Insert Items Detail 4. Submit Items 5. Submit Booking	1. Item Name: Iron 2. Quantity: 500 3. Total Weight: 2000 4. Category: Raw Material	1. Create new item and booking to database 2. Show "Successful" message and redirect to agenthome.jsp	1. Create new item and booking to database 2. Show "Successful" message and redirect to agenthome.jsp	Pass Pass
T13.2	Apply Booking which weight over the ship remaining capacity	1. Select Schedule 2. Select 3. Insert Items Detail 4. Submit Items 5. Submit Booking	1. Item Name: Iron 2. Quantity: 500 3. Total Weight: 10000 4. Category: Raw Material	1. Show "The ship is not enough capacity" Message	1. Show "The ship is not enough capacity" Message	Pass

Test ID: 14

Test Description: View Booking's Status

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T14.1	View Booking's Status	1. Press "Bookings" link which at the top		1. Show all the Bookings and status (Pending, Accepted, Declined)	1. Show all the Bookings and status (Pending, Accepted, Declined)	Pass

Test ID: 15

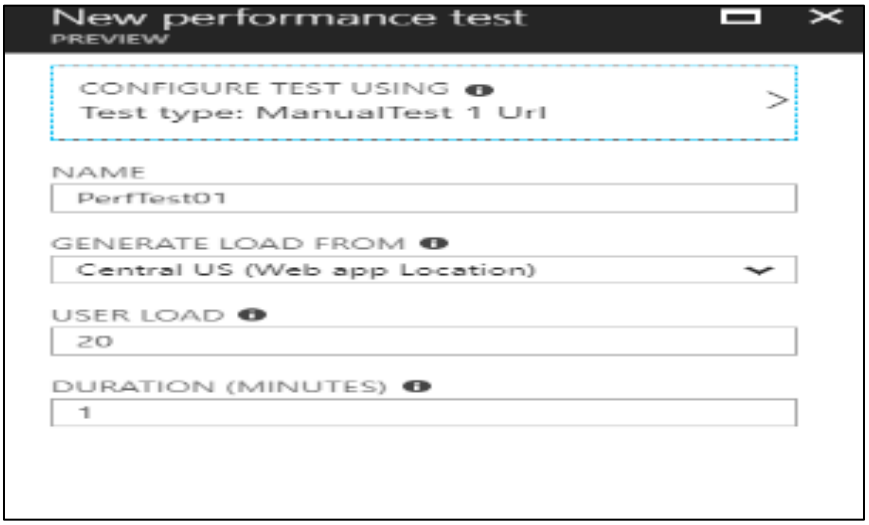
Test Description: View Customers

Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T15.1	View Registered Customers	1. Press "Customers" link which at the top nav-bar		1. Redirect to the agentcustomer.jsp	1. Redirect to the agentcustomer.jsp	Pass

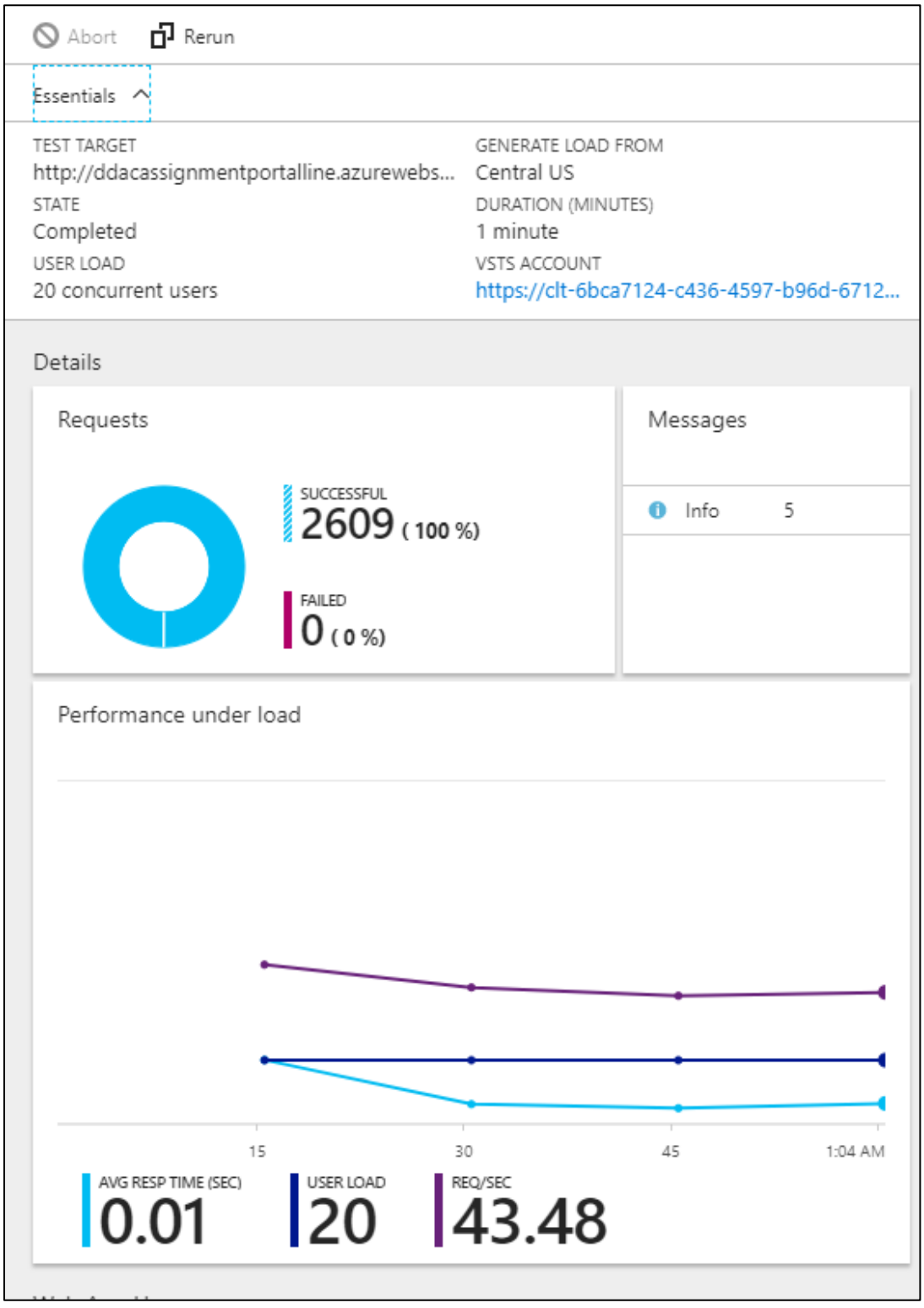
Test ID: 16						
Test Description: Add New Customer						
Test ID	Test Target (Function)	Test Steps	Data Input	Expected Result	Actual Result	Status(Pass/Fail)
T16.1	Add Customer by inserting correct information in all editable field	1. Press "Add New Customer" button which in page. 2. Enter Input 3. Submit the form	1. Name: Tan Li Feng 2. IC Number: 960411017113 3. Nation: Malaysia 4. Gentle: Male 5. Contact: 0167364254 6. Address: 6,Jalan Bukit Impian 12	1. Customer Account is created successfully in database. 2. Show succesful message and redirect user to agenthome.jsp	1. Customer Account is created successfully in database. 2. Show succesful message and redirect user to agenthome.jsp	Pass Pass
T16.2	Add Customer by inserting wrong data type and leave some blanks in some editable field.	1. Press "Add New Customer" button which in page. 2. Enter Input 3. Submit the	1. Name: Tan Li Feng 2. IC Number: NULL 3. Nation: Malaysia 4. Gentle: Male 5. Contact: 0167364254abc 6. Address: 6,Jalan Bukit Impian 12	1. Show "Please enter proper data type in state box" message 2. Show "Please make sure all information are inserted" message	1. Show "Please enter proper data type in state box" message 2. Show "Please make sure all information are inserted" message	Pass Pass

5.2 Load Performance Testing

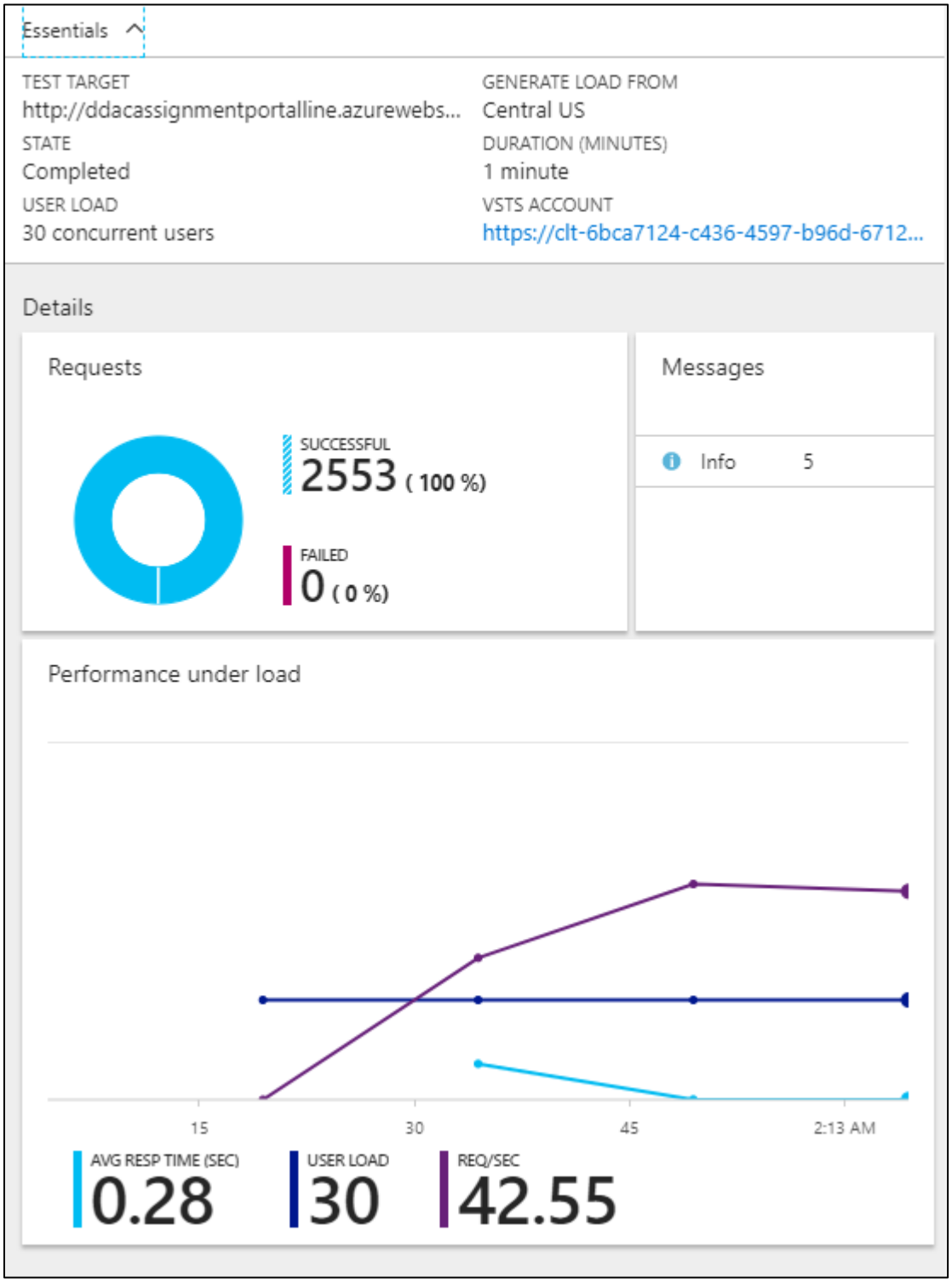
Performance is mostly about response time of the web application. This response time should be in acceptable intervals, and should not increase if transaction count increases (Basaraner, 2013). Performance testing is done with the help of the functionality provided as part of the Azure web application. The test is run on the main web application resource and the getting increase the users incrementally. Each test tests 20 to 40 user load for 1 minute. Results collected include response time and failed request.



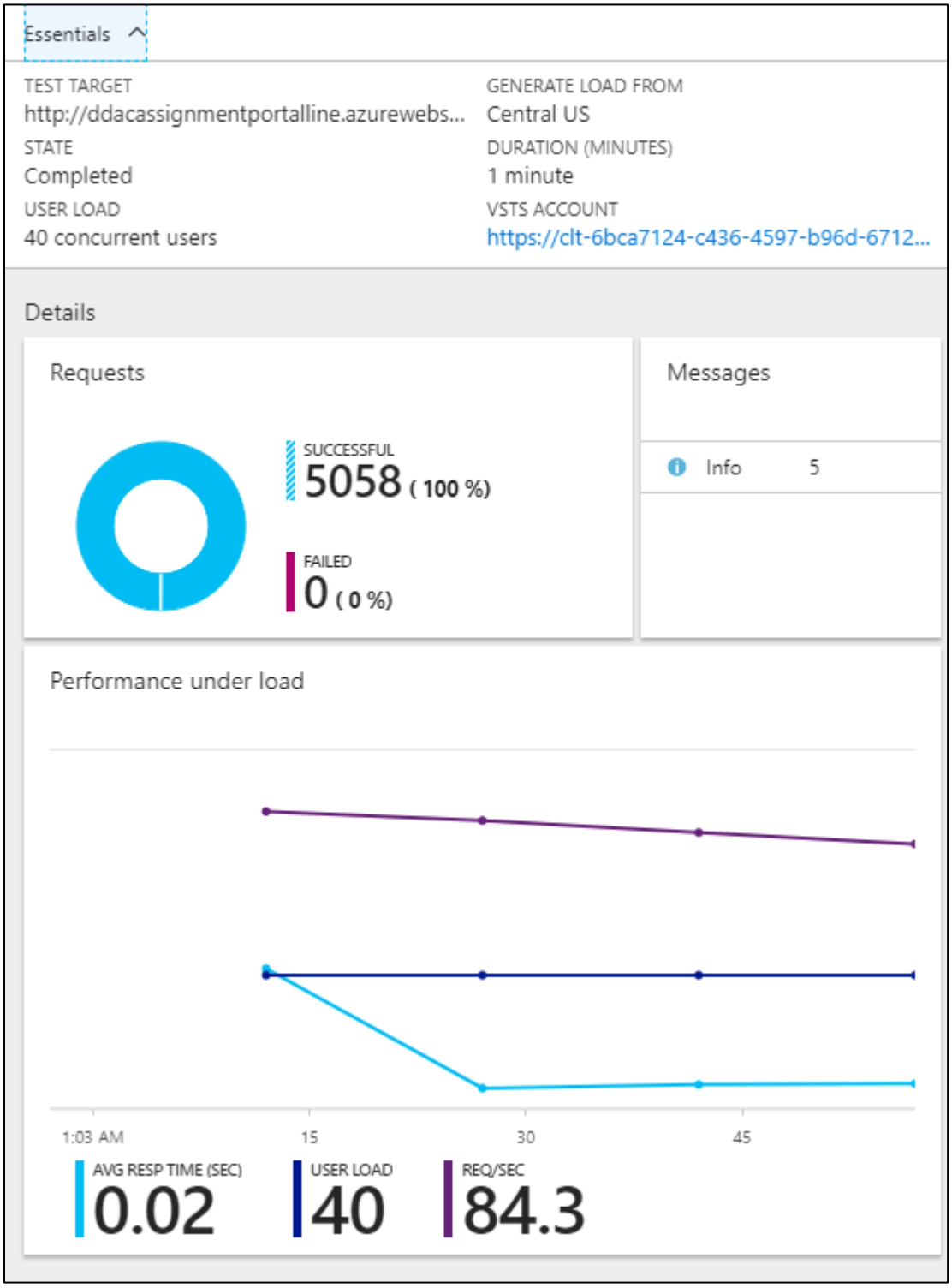
5.2.1 Result of 20 User Load



5.2.2 Result of 30 User Load



5.2.3 Result of 40 User Load



5.2.4 Analysis

This is based on the above load test chart provided by the developer. The Maersk Line Container Management System is a high performance for handling a large number of users in 60 seconds.

In this load test, the developer provides two scenarios. Two scenarios explaining the diagram which a scenario to handle 20 users in 1 minute, another scenario to handle 40 users in 1 minute.

In the first scenario, in the web application that handles 20 users in 1 minute, the load test shows that the performance of the web application is performed well without suspense, and the web application takes average 0.1 second for the response time per page display to each user.

In the second scenario, the web application was handling 40 users in 1 minute, the load test indicates that the performance of the web application is good and the web application has a response time of 0.2 seconds per page display by the user Indicates that.

After comparing these two scenarios, developers are analyzing that the Maersk Line Container Management System is able to deliver and maintain high performance for a medium scale of concurrent users browsing to the application within 60 seconds.

6.0 Conclusion

In Conclusion, I am appreciated that the proposed system is carried out successfully and which is able to achieve all the deliverables that defined in before but there are still having shortage within the system. In the last few stage of the development, the trial plan of Azure Account has expired which cause the load performance testing can't be conducted by testing heavier load of users. I am also appreciated to be given an opportunity to finish this project with a proper cloud design pattern and cloud application development, especially this assignment is being brought into case which similar as a real-life project. I had learnt a lot from this project such as proper programming techniques and project management skills which are useful in my future employment. For the challenging facing during this assignment might be doing the brainstorming about the design of GUI and research for the related source code. After finished this project, I am really get learnt about how a program is designed in patterns properly to lead the system become high cohesion and low coupled.

References

Basaraner, C., 2013. *10 Software Quality Factors That Should Always Be Remembered.* [Online]
Available at: https://dzone.com/articles/10-groups-software-quality
[Accessed 11 4 2018].
Bootstrap, 2018. *Bootstrap.* [Online]
Available at: https://getbootstrap.com/
[Accessed 11 4 2018].
Cavale, A. B. R. J. &. K. J., 2017. *Scale instance count manually or automatically..* [Online]
Available at: https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/insightshow-to-scale
[Accessed 11 4 2018].
Microsoft, 2018. *Choose a cloud SQL Server option: Azure SQL (PaaS) Database or SQL Server on Azure VMs (IaaS).* [Online]
Available at: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-paas-vs-sql-server-iaas
[Accessed 11 4 2018].
Mozilla, 2018. *JavaScript.* [Online]
Available at: https://developer.mozilla.org/bm/docs/Web/JavaScript
[Accessed 11 4 2018].
Yard, E., 2018. *Top 10 Advantages of Platform as a Service.* [Online]
Available at: http://www.engineyard.com/whitepapers/top-10-advantages-of-platform-as-aservic
[Accessed 11 4 2018].

Appendix

Project Links

More information for the system development and demonstration:

GitHub Link: https://github.com/lifeng1996/DDACportal

Azure Link: https://ddacassignmentportalline.azurewebsites.net/DDACPortal/

Microsoft Stream Link:

https://web.microsoftstream.com/video/216c6811-0a1e-4d29-8b28-4fe88970af74

Test Account for demostration

Admin Account

Username: test

Password: test

Agent Account

Username: testagent

Password: test