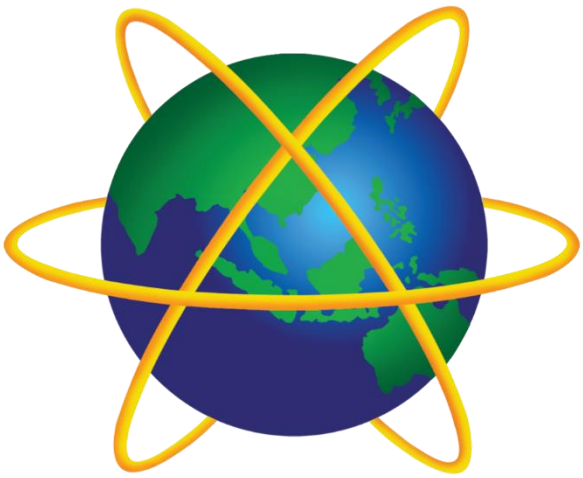


ASIA PACIFIC UNIVERSITY OF TECHNOLOGY & INNOVATION

TECHNOLOGY PARK MALAYSIA

INDIVIDUAL ASSIGNMENT

CT071-3-3-DDAC

DESIGNING AND DEVELOPING APPLICATION ON THE CLOUD

Name: Lew Wai Keat

TP Number: TP031801

Lecturer: DR. KALAI ANAND A/L RATNAM

Intake Code: UC3F1702(IT)ISS

Hand-in date: 20th November 2017

Contents

Introduction

Project Background

Ukraine International Airlines (UIA) is a flagship carrier and largest airline company in Ukraine. It operates both domestic and international flights and cargo services to Europe, Middle East, America and Asia. UIA has long used technology to reduce costs, innovate, and improve customer service. It has gone to a paperless cockpit and uses sophisticated software for analyzing fuel economy. However, the site experienced severe denial-of-service (DOS) attacks, which hurt site performance and reliability, and it did not have the performance needed to host visitors from many parts of the world.

Dmitriy Prudnikov, Chief Information Officer at Ukraine International Airlines is looking to develop Online Flight Booking System that uses public cloud to solve all these problems. Decision was made on choosing Azure for the cloud service of its website. The UIA web need to ensure reliability, availability and performance during the expansion of their market, therefore Azure could services is the fitting choice for UIA.

Objective

Objective of this project UIA Online Flight Booking System are:

- Provide a user-friendly system
- Develop a system that has high performance
- Provide a system with high availability

Project Deliverables

UIA Online Flight Booking System has the ability to be accessed through internet browsers and the system aims to deliver to all potential customers both domestically and international. The system provides functions such as:

- Customer Login
- Search of flights through destination
- Listing available flights
- Booking of flight seats
- View booked flights

Requirement Specification

a) **Performance**: Able to maintain website the fluidity during peak hours
b) **Maintainability**: Able to maintain web access during website maintenance
c) **Monitoring**: Able to monitor web app performance and identify possible problems on the go
d) **Availability:** Able to provide web app services to users whenever and wherever
e) **Scalability:** Able to scale up or down to meet the demand users access at any given time

Summary of major function

There are total 2 users and each of the users has different functionalities as shown as following:

i. Customer

- Register Account
- Login into own Account
- View and edit own customer profile
- Searching Flight/Destination
- Make booking flight
- Make Payment
- Request to cancel

ii. Administration/Staffs

- Manage and view booking details
- Update Customers Details
- Flight cancellation
- Received Payment
- Generate Receipt

Project Plan

Project plan Design

Ukraine International Airlines (UIA) website is required to be published onto Microsoft Azure. Azure is chosen because of the reliability and scalability. UIA allow customers to search and book for available flight based on their destination through the website presentable user interface. The Chief Information Officer has identify the current website is having performance and reliability issues due to overwhelm of users accessing the website, thus the website need Azure cloud service is the solution to the problem where its always available and reliable. Furthermore, resources can be scale up or down to save cost and means the users demand at any given time.

Task Name	Duration
Planning on the Design of web application for UIA	10 Days
Cloud Pattern Selection	5 Days
Architecture Drawing	5 Days
Web Design with UI template by using Mockup	10 Days
Implement Code and design Web	15 Days
Azure Configurations	5 Days
Test Plan Design	1 Day
User test and Load Test	3 Days
Correction from the code fail based on the test plan	3 Days
Technical Documentation	7 Days

Table 1 Project Plan Design

Gantt Chart

Task Mode	WBS	Task Name	Duration	Start	Finish
	1	Introduction	5 days	Tue 6/2/18	Mon 12/2/18
	1.1	Project Background	1 day	Wed 7/2/18	Wed 7/2/18
	1.2	Objectives	1 day	Thu 8/2/18	Thu 8/2/18
	1.3	Scope	1 day	Thu 8/2/18	Thu 8/2/18
	1.4	Requirement Specification	1 day	Fri 9/2/18	Fri 9/2/18
	1.5	Summary of Major Functions	1 day	Sat 10/2/18	Sat 10/2/18
	2	Project Plan	2 days	Mon 12/2/18	Tue 13/2/18
	3	Design	7 days	Wed 14/2/18	Thu 22/2/18
	3.1	Architecture	4 days	Fri 23/2/18	Wed 28/2/18
	3.1.1	Assumption	1 day	Thu 1/3/18	Thu 1/3/18
	3.1.2	List of Azure Services Identification	1 day	Fri 2/3/18	Fri 2/3/18
	3.1.3	Architecture Diagram	2 days	Mon 5/3/18	Tue 6/3/18
	3.2	Application modeling	3 days	Thu 8/3/18	Mon 12/3/18
	3.2.1	Use Case	1 day	Mon 12/3/18	Mon 12/3/18
	3.2.2	Class Diagram	1 day	Tue 13/3/18	Tue 13/3/18
	3.2.3	Sequance Diagram	1 day	Thu 15/3/18	Thu 15/3/18
	4	Implementation	57 days	Thu 8/2/18	Fri 27/4/18
	4.1	Implementation on website	57 days	Thu 8/2/18	Fri 27/4/18
	4.2	Implementation on documentation	7 days	Fri 27/4/18	Sun 6/5/18
	4.2.1	Publishing Application to Azure	1 day	Mon 7/5/18	Mon 7/5/18
	4.2.2	Publishing Database to Azure	1 day	Tue 8/5/18	Tue 8/5/18
	4.2.3	Traffic Manager	2 days	Wed 9/5/18	Thu 10/5/18
	4.2.4	Screenshot of the Application	2 days	Fri 11/5/18	Mon 14/5/18
	5	Test Plan & Testing Discussion	7 days	Mon 14/5/18	Tue 22/5/18
	6	Conclusion	1 day	Wed 23/5/18	Wed 23/5/18
	7	References	1 day	Thu 24/5/18	Thu 24/5/18
	8	Project Closed	78 days	Tue 6/2/18	Thu 24/5/18

Figure 1: Grant Chart

Figure 2: Grant Chart Flow

Design

Architecture

List of Azure services identification

- **Pricing Tier**: S2 Standard (50 DTUs)
- **Web App**: Web App Microsoft.
- **SQL Database Server**: brendenlew.database.windows.net

Cloud Architecture

Retry Pattern enable the web app to handle transient failures, it will tires to connect to a services or network resources by transparently retrying a failed operation. This can improve Stability of the UIA web.

As shown in figure above, retry pattern invokes an operation in a hosted service. It helps handle failures if an

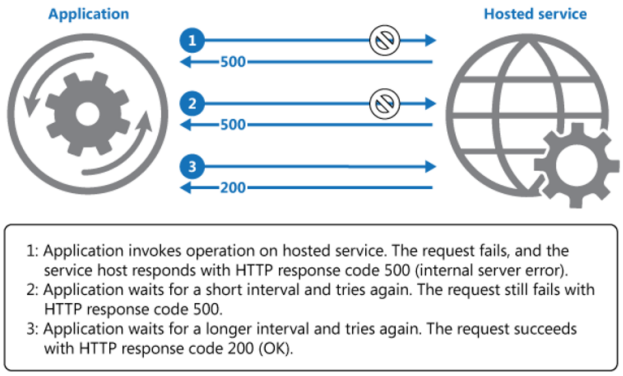


Figure 3: Retry Pattern

application fail is detected with the use of following solutions:

- **Cancel:** The application will cancel the operation and report an exception if it had determined that the failure is unlikely to be successful if repeated.
- **Retry:** The application will retry the failing request if the fault is reported unusual and might have been caused by rare circumstances.
- **Retry after delay:** The application will wait for a better time and retry the request if the fault is caused by commonplace connectivity or busy failures.

Retry Pattern is recommended when an application experience temporary faults with a remote services, the chances of success increases by repeating the previous failed request on multiple attempts.

2.1.1. Architecture Diagram

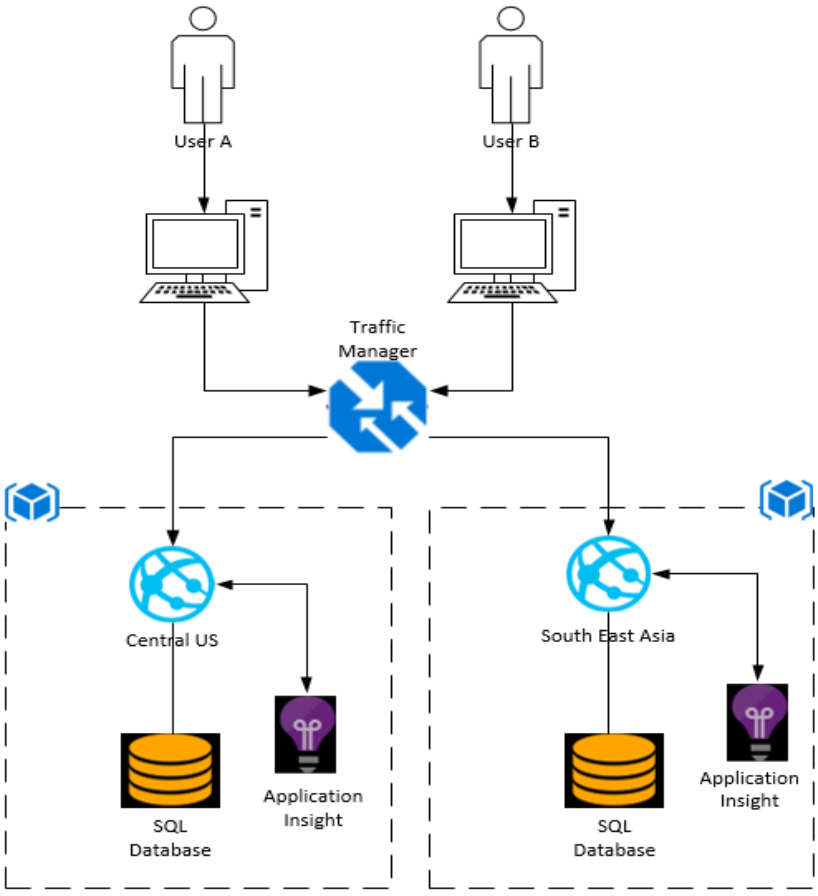


Figure 4: Architectural Diagram

Application Modeling

Use Case

The use case diagram is created to support developer in visualizing the fundamental requirement of UIA booking system. It also provide clear view of relationship interaction between the users and system. Use case diagram generally show group of use cases of use cases for the complete system or breakout a group of use cases with related functionality.

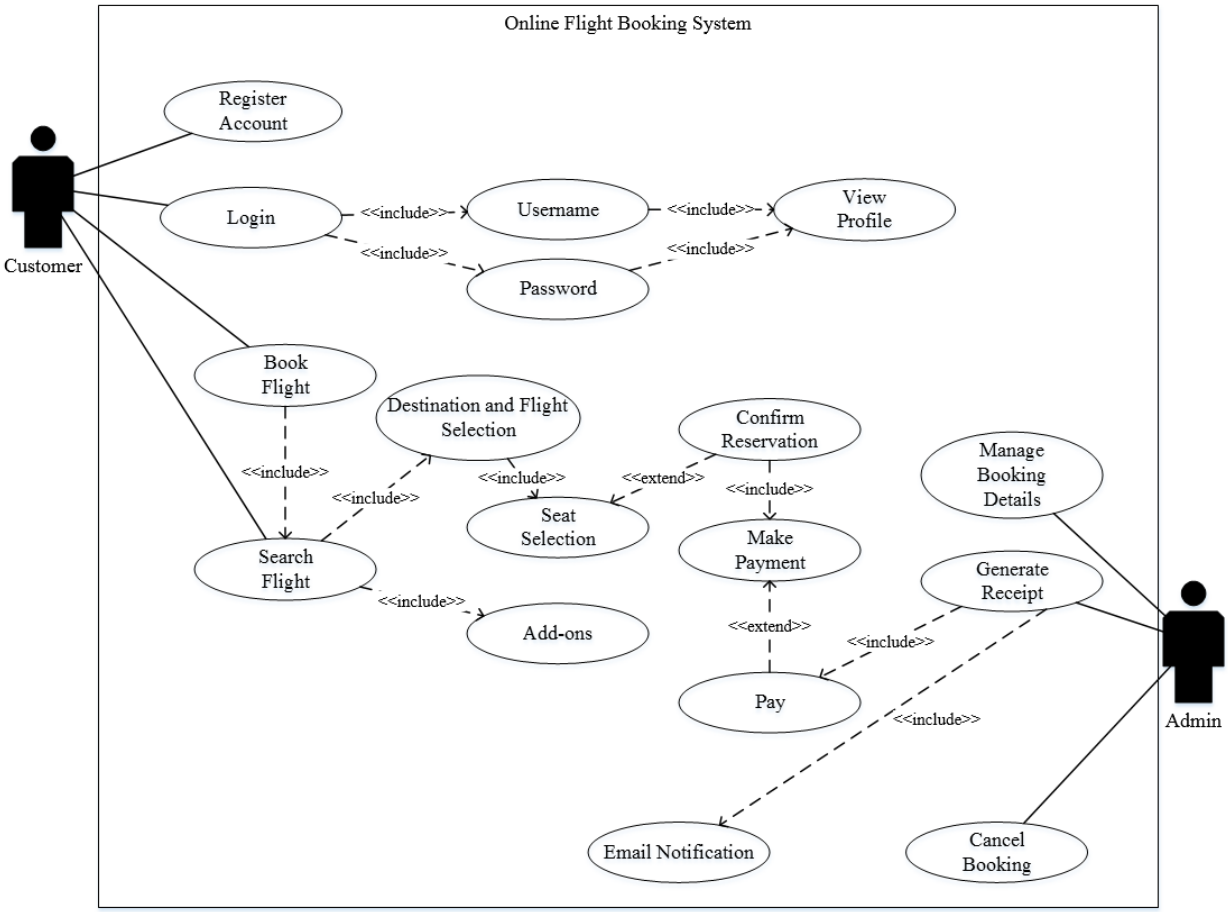


Figure 5: Flight Booking Use Case

Sequence Diagram

Sequence diagram is used to describe the flow of messages, actions, and events between objects or demonstrates parallel initiations and processes. It a collaboration of objects based on a time sequence that show how the object is interacting in a given situation.

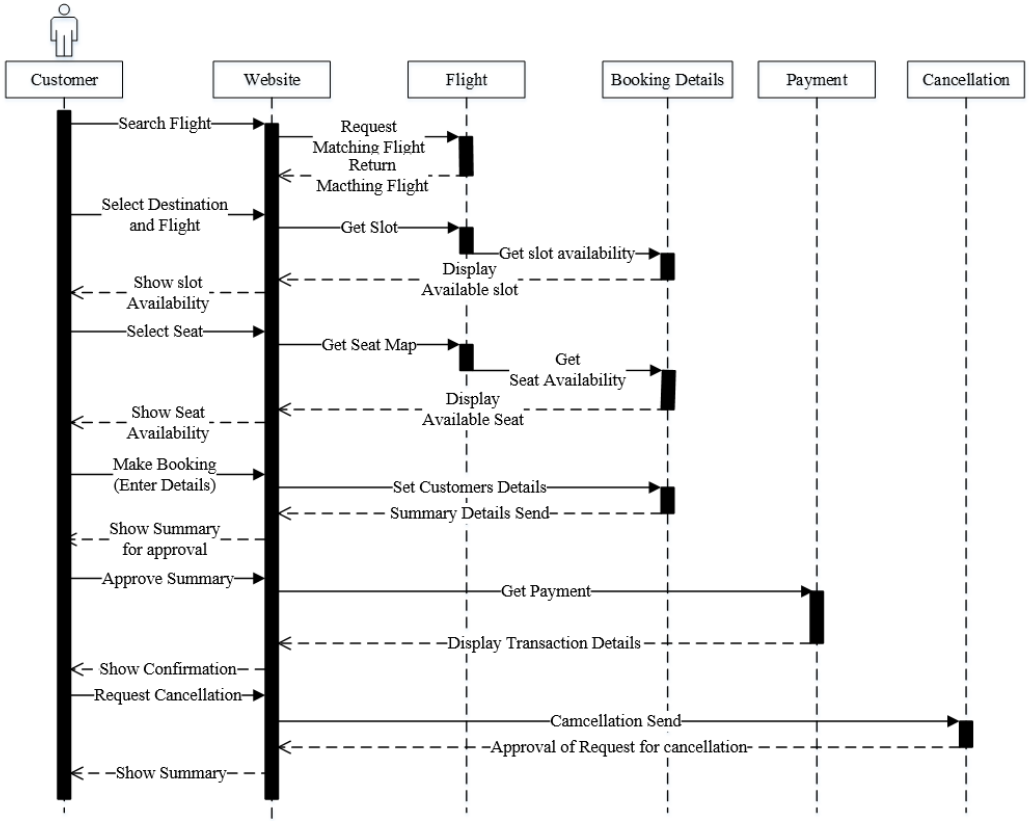


Figure 6: Sequence Diagram for Booking Flight

Figure above shows, how a customer can interact with the flight booking system.

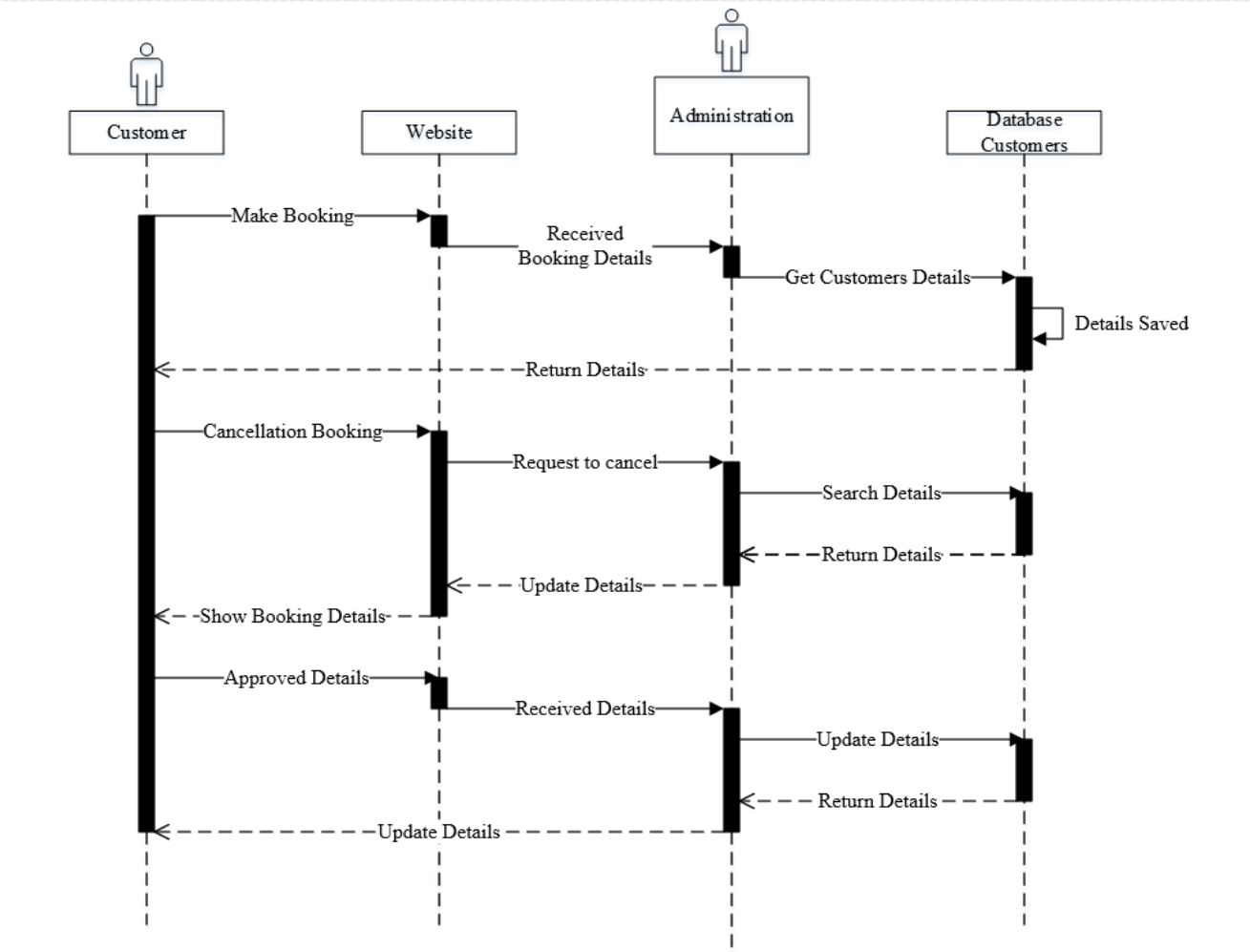


Figure 7: Sequence Diagram of Administration

Figure above shows how a admin can update and edit the details of the customers flight booking schedules.

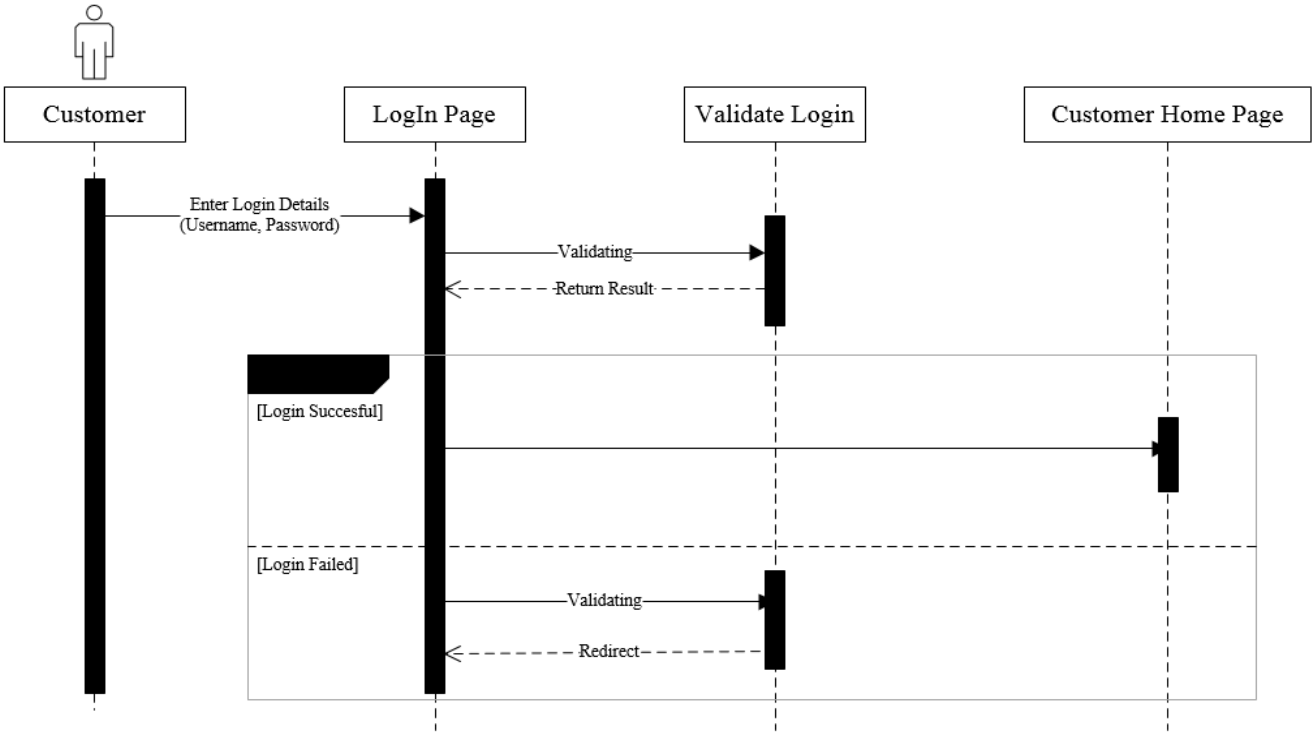


Figure 9: Sequence Diagram Login

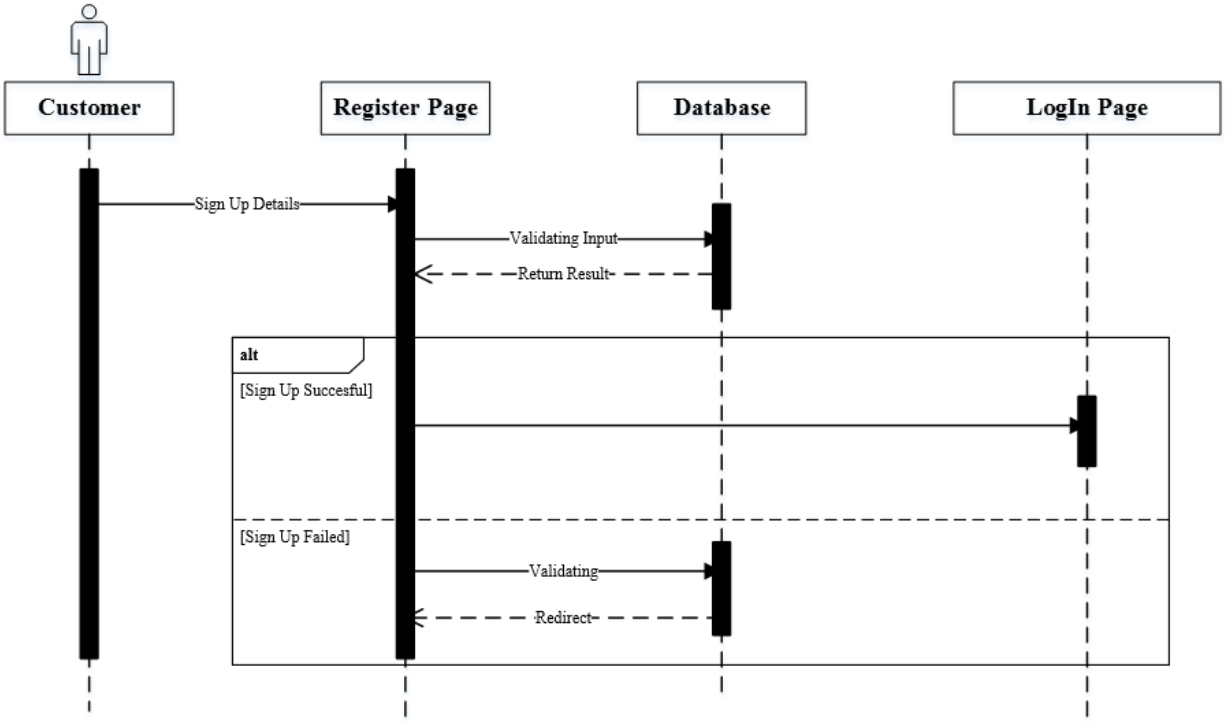


Figure 8: Sequence Diagram Register

Implementation

Publishing Application to Azure

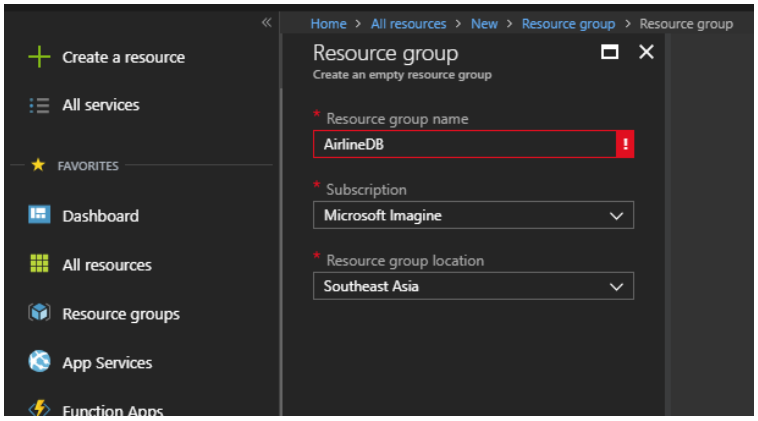


Figure 10: Create new resource group

Developer is required to create a resource group with a unique name (AirlineDB) and select location at Southeast Asia as Shown in Figure above. A resource group is used to keep related resources in a group, so it's more easily accessible.

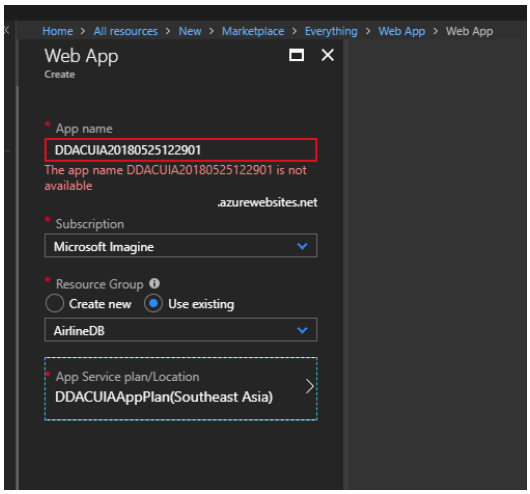


Figure 11: Create Web App

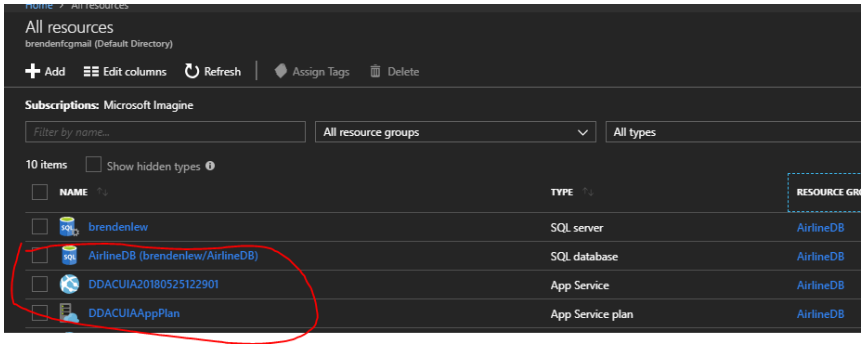


Figure 12: Create App Service

Above shown how to create web app in Microsoft Azure portal as well as the resources created for UIA web app.

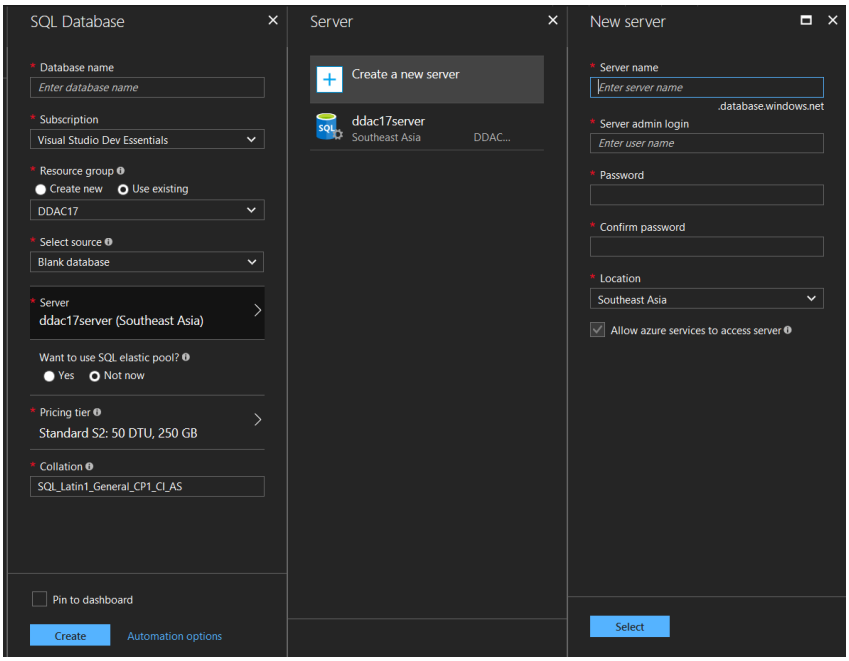


Figure 13: Create new database and server

After successfully created resource group, Web App and SQL database inside the resource group. Free App Service plan was selected for this assessment. It can be scale up or down later on. The Web App support both Windows and Linux Platform With built in auto scale and load balancing that can improve the performance and stability of the Application. Furthermore, the UIA Web app database is hosted by Azure with also provide the scalability which UIA can later scale up whenever deem needed. The SQL database created with the following admin Login and Password:

- Server admin name : brendenlew
- Password : P@SSW0RD
- done created web app and SQL database in resource group, developer open Visual Studio
- 2017 and selected ddac17 project (UIA Booking System) and click on Publish. In the Publish page, it will ask the developer to choose where to be publishing and developer has chosen an existing Microsoft Azure App Service which is “ddac17”. After successful published, the UIA Booking System hosted in Southeast Asia can be accessed with URL of http://ddac17.azurewebsites.net/.

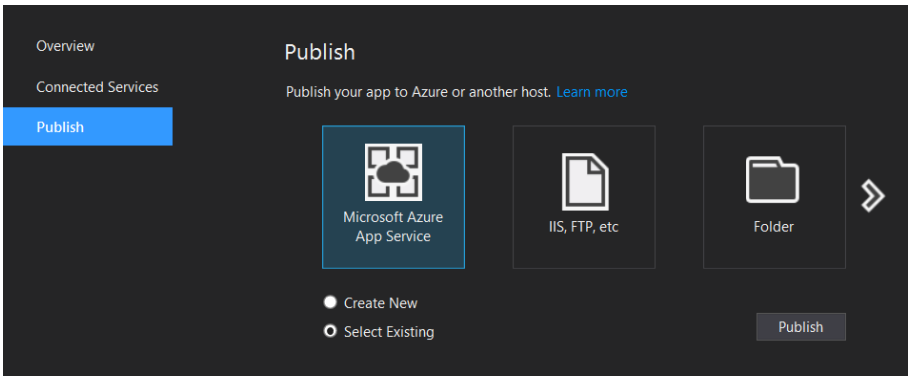


Figure 15: Publish Website booking Flight to cloud

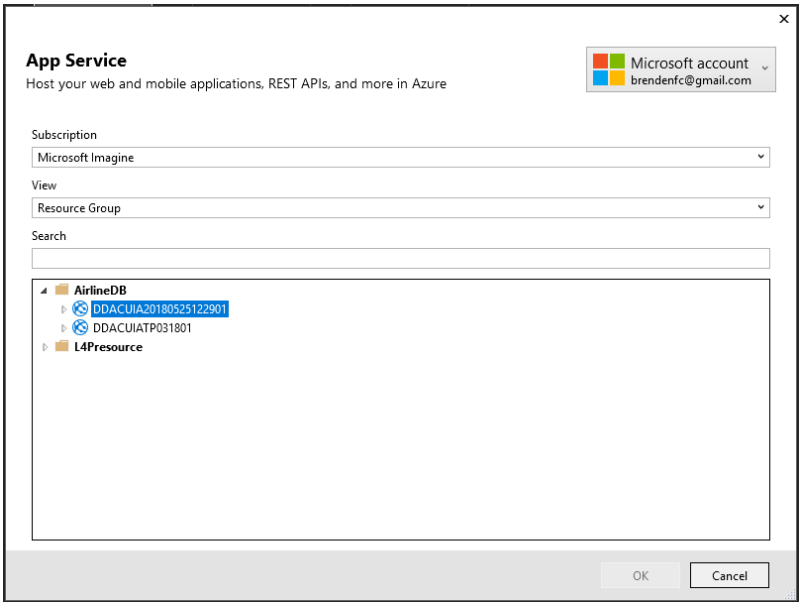


Figure 14: Deploy web app

To Publish the developed Web app, Visual Studio has integrated function that allow web app publishing to Azure. Developer is required to open the project in Visual Studio, select the project and click on Publish. In the Publish Page, it will ask to choose where for the app services that is created previously. After successfully published, the UIA Web App hosted in Southeast Asia can be accessed with URL of https://ddacuiatp031801.azurewebsites.net/

Publishing Database to Azure

With the system Published onto Azure, Database is required by the UIA Web App. Hence in order to manage the database, Microsoft SQL Server Management Tool (SSMS)2017 is used.

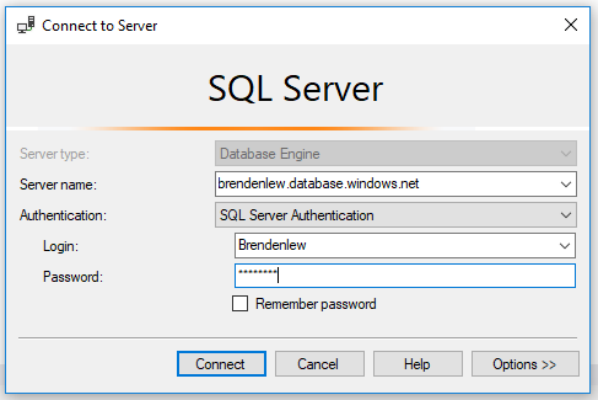


Figure 16: Accessing Azure Database

To access Azure SQL Database with SSMS the following information is required Server name , Login ID and Password is required as following:

- Server Name: brendenlew.database.windows.net
- Login : brendenlew
- Password : P@SSW0RD

Screenshot of the Application

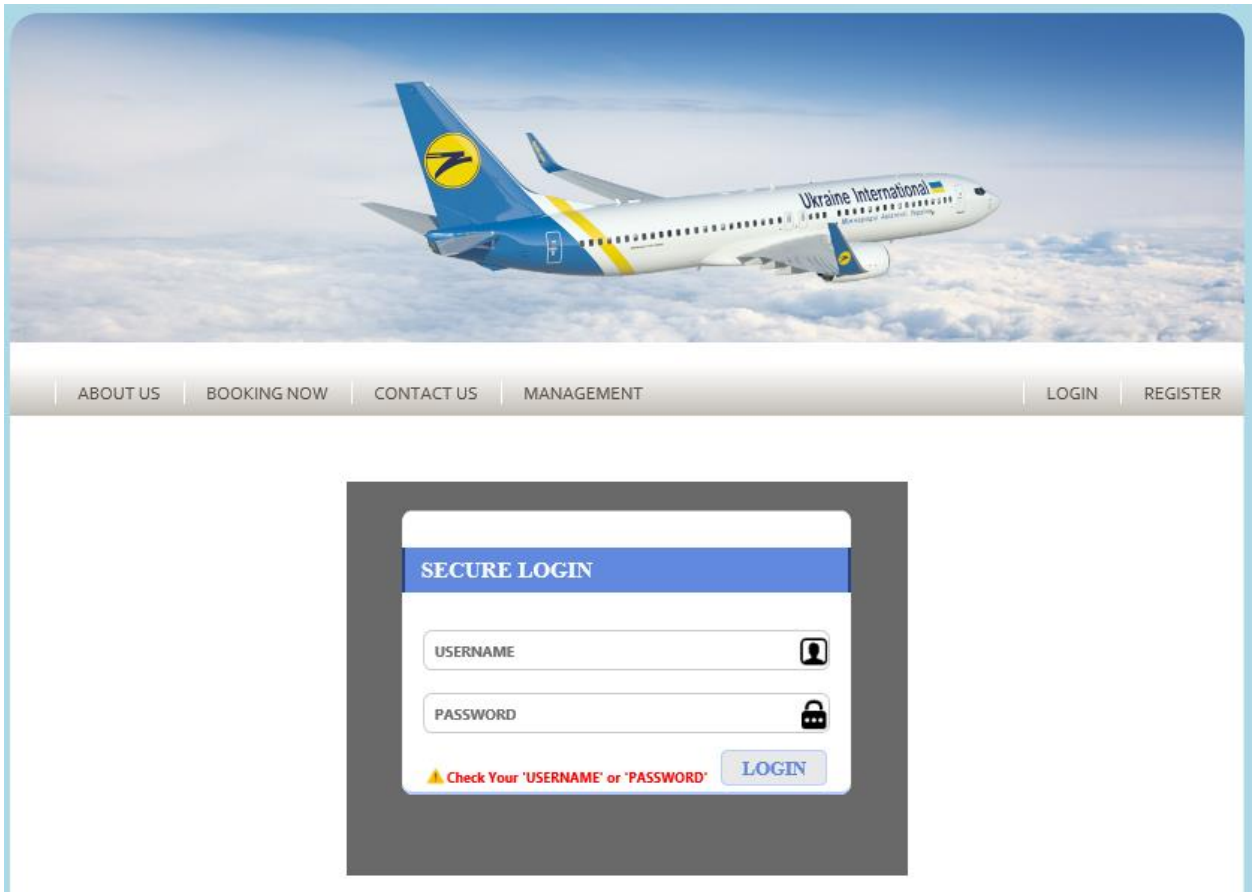


Figure 17: Login Page

The above figure is the main page which customer or admin is required to login with valid username and password before accessing the website content.

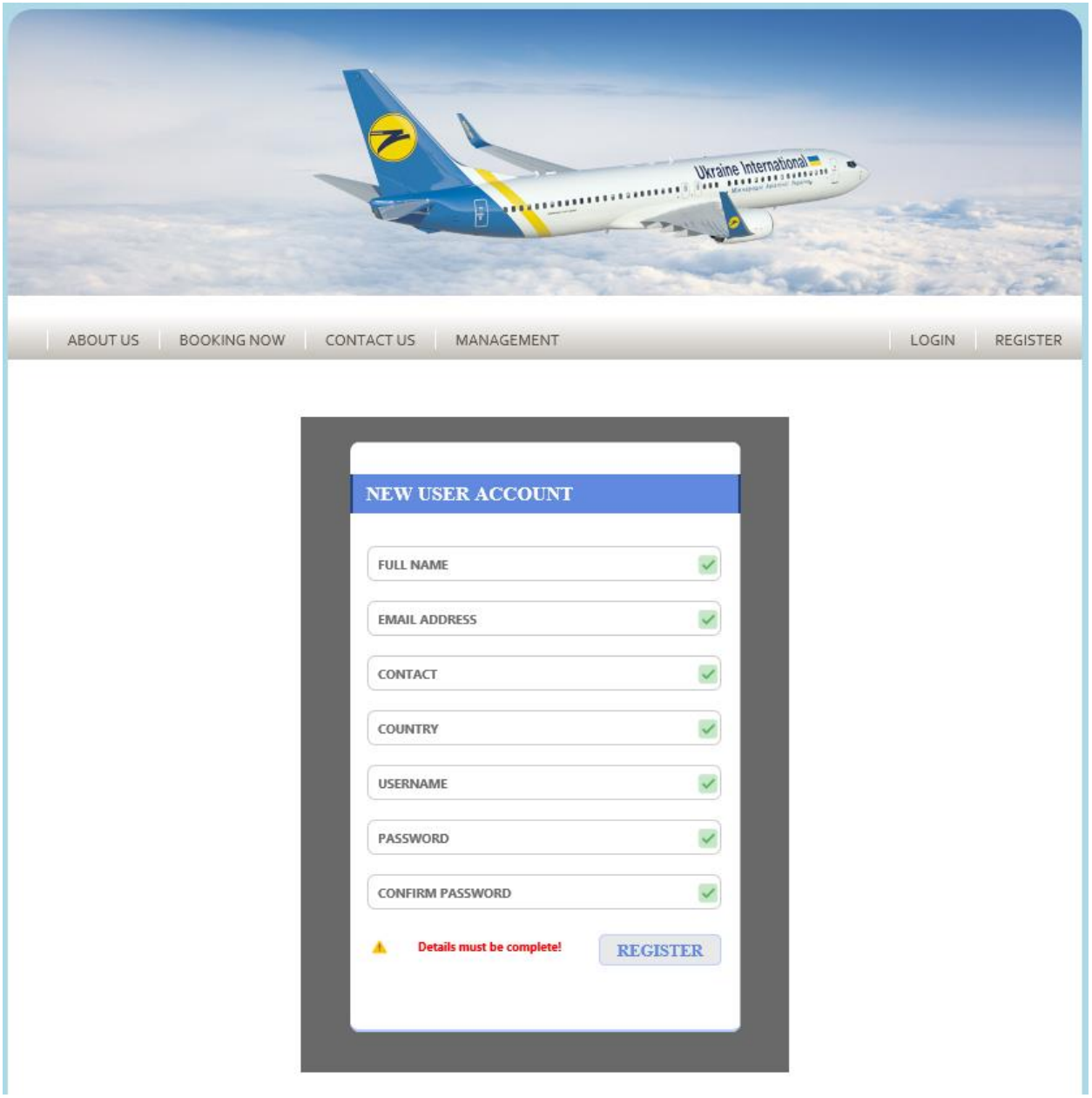


Figure 18: Registration Form

Above shows the registration page where new user is required to fill up the form for registration.

Upon successful register user can proceed to login with the username and password created.

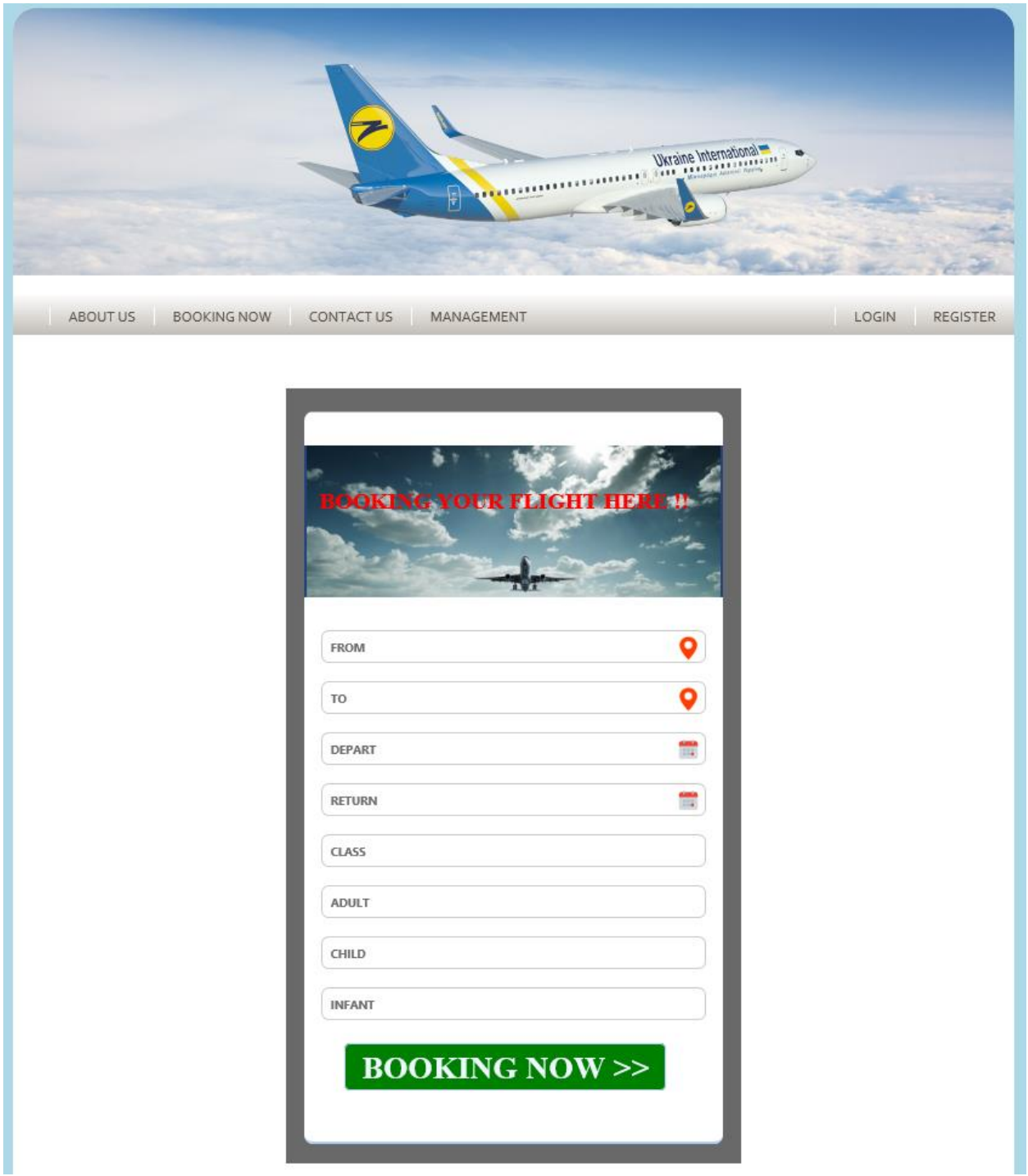


Figure 19: Booking Page

These page customers can book their flight ticket based on the needed.

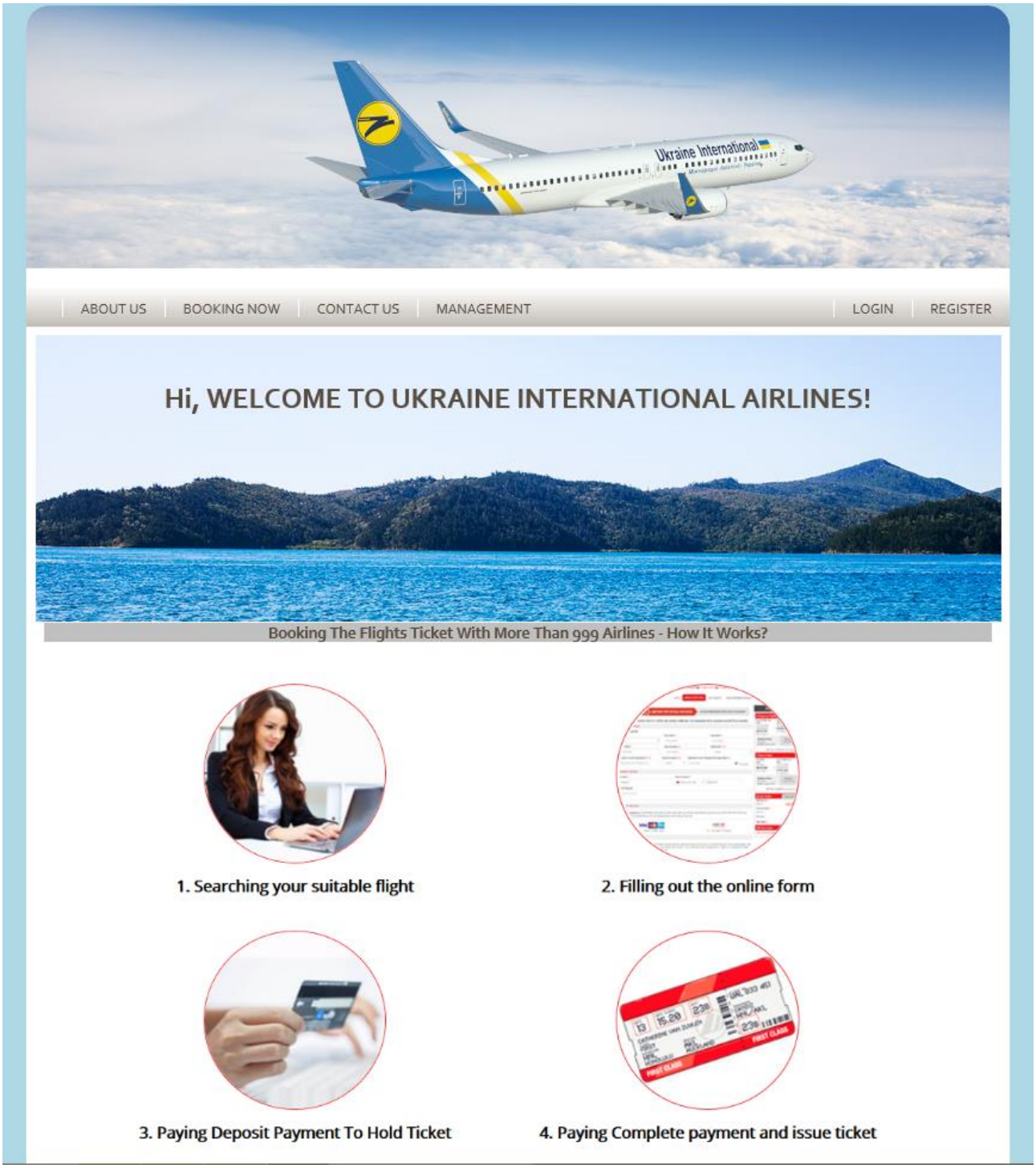


Figure 20: About Us page

This is the page where admin can control the database of the customers and flight details. They can edit, delete and update the details through this page.

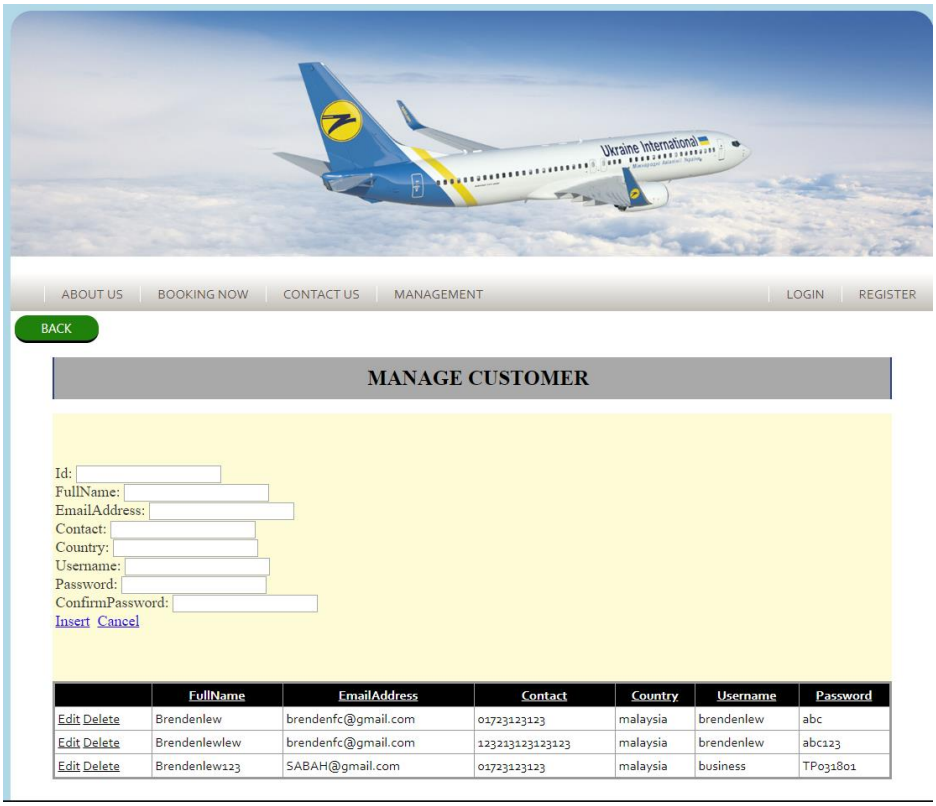


Figure 21: Manage Customer Detailes

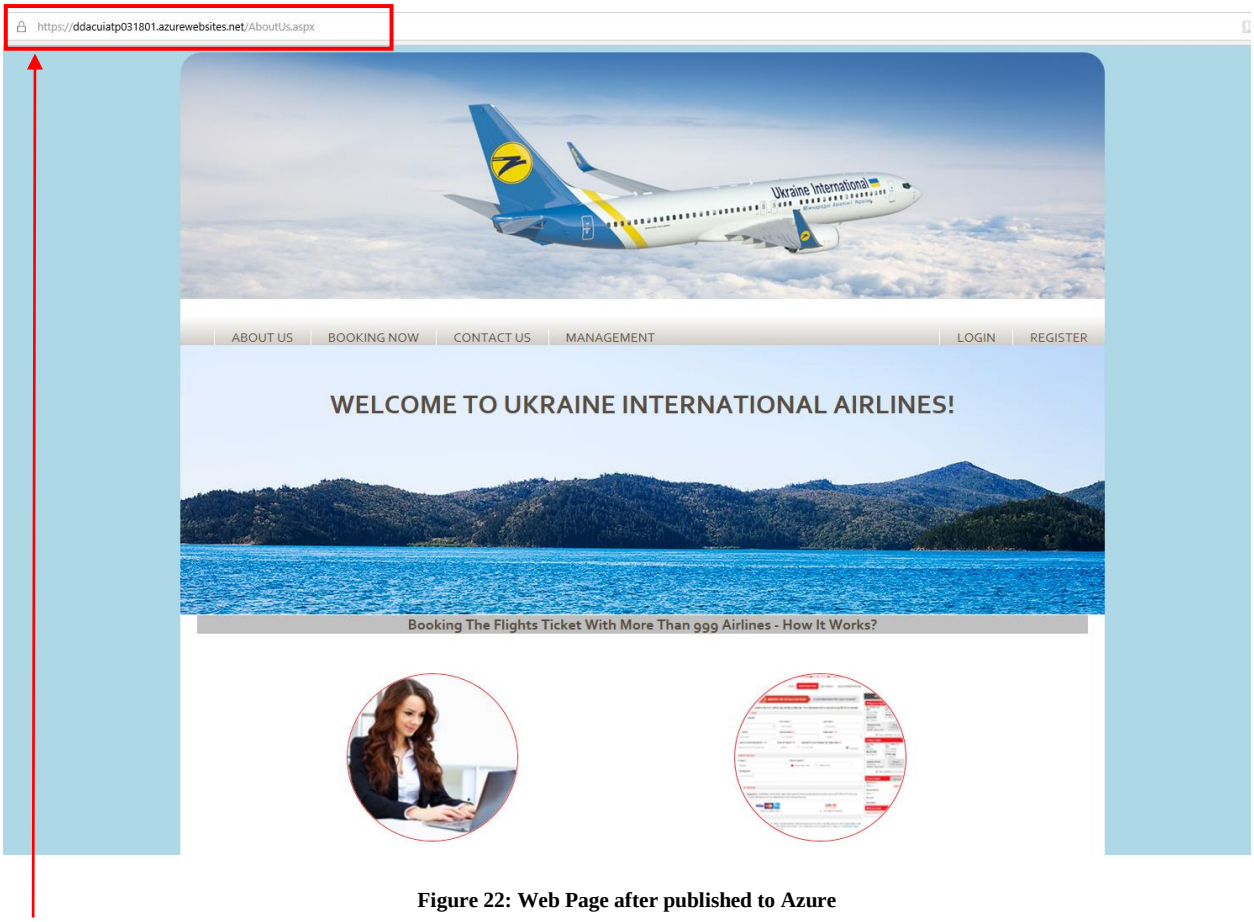


Figure 22: Web Page after published to Azure

In the link of the website, after publish in Azure account the link is https://ddacuiatp031801.azurewebsites.net/AboutUs.aspx

Test Plan & Testing Discussion

Test Plan

Table below shows the sample of unit testing for the UIA Flight Booking Web App that is hosted by Microsoft Azure.

Test Plan for Customer

Register

Test Case ID	Test Criteria	Expected message	Actual message
1	Input all the required fields	Successfully Registered	Save Password? -Direct to Profile Page of the user
2	Empty field left	Please fill in the details	As expected
3	After adding new user details, proceed to Login Page to verify the login.	Login Success	As expected

Login

Test Case	Username	Password	Expected message	Actual message
1	Empty	Empty	Username and password empty	There are some error occur during login
2	Brendenlew	Empty	Please enter password. Password is empty!	There are some error occur during login
3	Empty	abac	Please enter username. Username is empty!	There are some error occur during login
4	Brendenlew	abc	Login Success	As Expected

Booking Flight

Test Case	Test Criteria	Expected message	Actual message
1	Input all the booking details. Click on the button "Book Now"	Booking is Successfully added.	As Expected
2	The input is empty	Booking will not successfully insert to database.	As Expected

Make Payment

Test Case	Test Criteria	Expected message	Actual message
1	Input all the payment details. Then click "make payment" button.	You have made your Deposit! Thank You.	As Expected
2	If one of the details is not complete, validation text will appear.	Validation will appear in red colour for example "Email is required".	As Expected
3	If customer not making a booking flight, the payment process will not be proceed.	Cannot proceed to payment page.	As Expected
4	Payment have been made by customer, receipt of deposit will be show on the page.	Payment successfully made by customer.	As Expected

Test Plan for Admin

Login

Test Case	Username	Password	Expected message	Actual message
1	Empty	Empty	Username and password empty	There are some error occur during login
2	Admin	Empty	Please enter password. Password is empty!	There are some error occur during login
3	Empty	123	Please enter username. Username is empty!	There are some error occur during login
4	Admin	123	Login Success	As Expected

Management

Test Case	Test Criteria	Expected message	Actual message
1	Edit/Update/Delete • type of aeroplane • aeroplane details • seat details • customer details	Details of criteria were successfully delete/update/delete.	As Expected
2	Generate Receipt • Receipt has been give to customer	Receipt successfully generates.	As Expected

Performance Testing

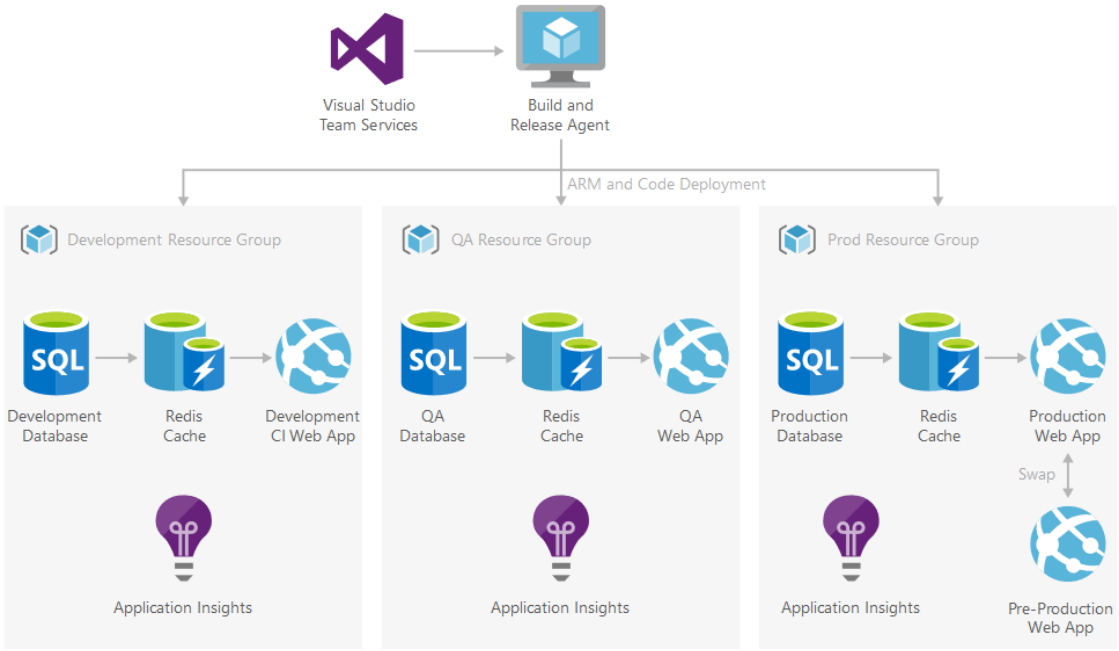


Figure 23: Testing flow for Azure

Load Testing is a technique used to examine the behavior of a system when subject to extreme demand or load is placed on it. Load testing is generally helps to identify the maximum capacity of an application or system as well as any bottlenecking (Techopedia.com, 2018).

Azure provide load testing tool for developer to place intents load on the web application and determine how many concurrent users can the web app handle. The results is then correspond to real time scenario to deem whether UIA web app is required to scale up or down.

The UIA Flight Booking Web App that is located in Southeast Asia will be tested first with App Service Plan of Standard (S2) with 250 user load in 5 minutes.

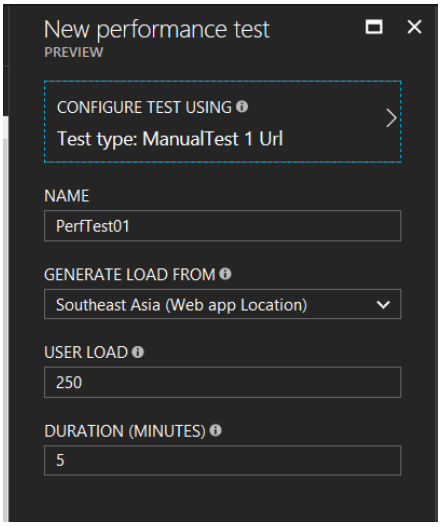


Figure 24: Adding first performance test

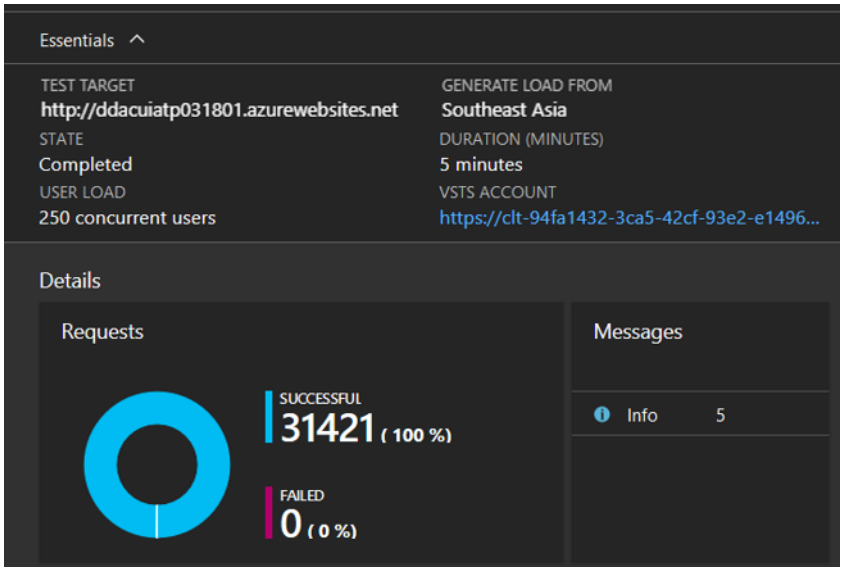


Figure 25: Details of Test Result

As figure above, developer testing the system using user load of 250 and duration of 5 minutes and click on Run Test to start performing the test of the system and wait about 10 minutes for the test to be done. The results of the test are shown below.

As figure above, the performance has 100% successful when having 250 users load in 5 minutes for Standard (S2) as Service Plan as it can withstand the load.

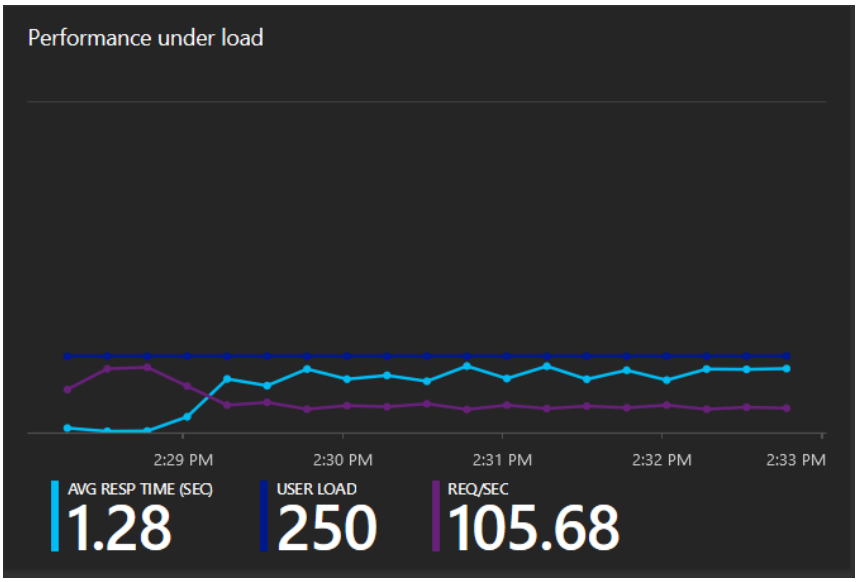


Figure 26: Performance under load

It has average response time of 1.28 in seconds with user load of 250. The Web App Usage is shown in figure below.

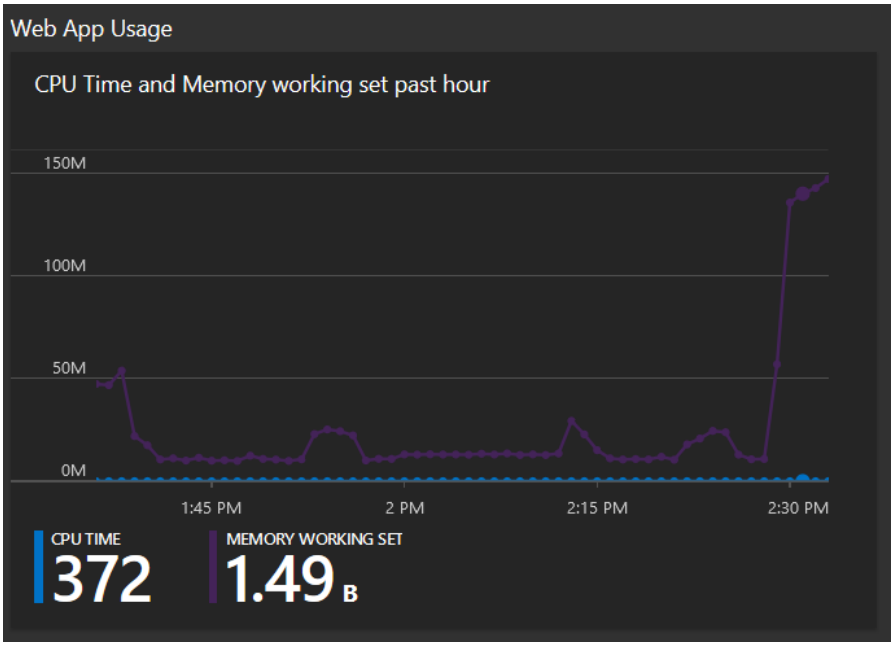
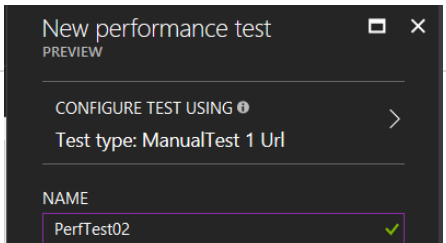


Figure 27 Web App Usage

It has shown that Standard (S2) App Service Plan is sufficient for the system to handle 250 users in 5 minutes, hence scaling up the Service Plan is needed. Next, the web app has scaled up to Standard (S2) with handle 1000 users in 5 minutes to perform testing.



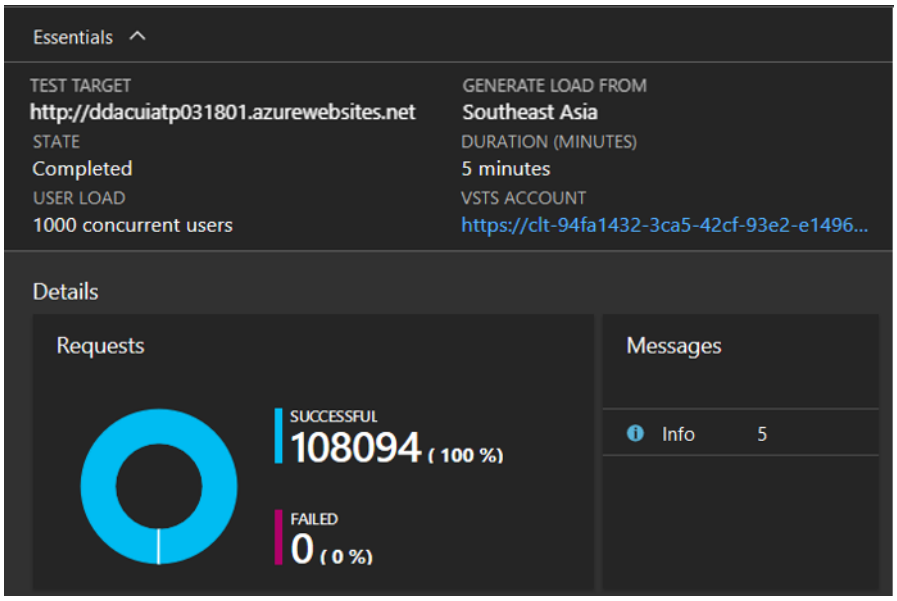


Figure 29: Details of Result

Add another new performance test with the same user load and duration in minutes as previous testing. Click on Run Test to start the test. Results of the test are shown in diagrams below.

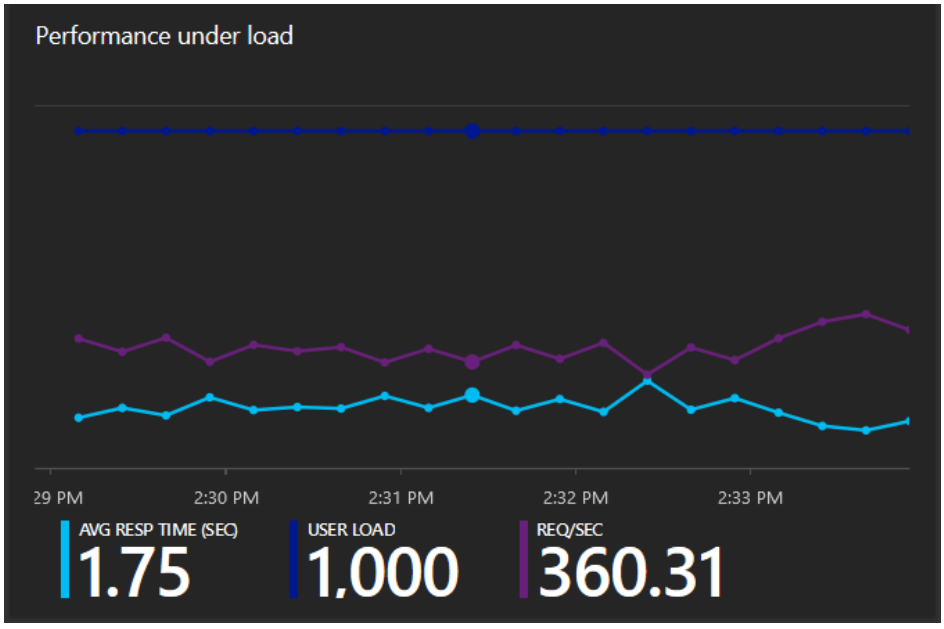


Figure 30: Performance under load

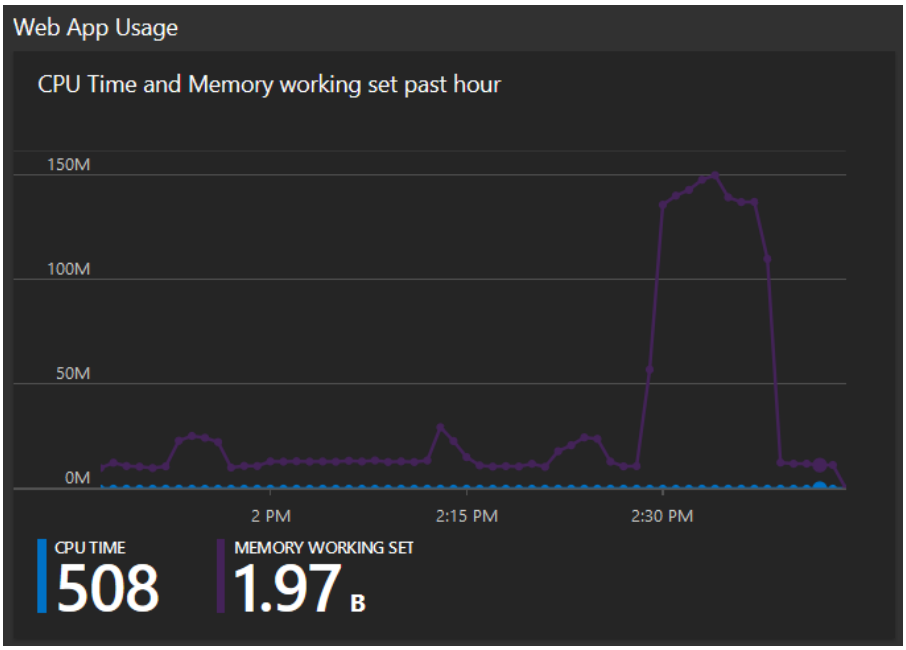


Figure 31 Web App Usage

It has shown that Standard (S2) as Service Plan is better as it has 100% success when 1000 users access the web concurrently in 5 minutes. It also has an average response time of 1.75 in seconds. Hence, it is better for Online Website UIA Booking Flight Ticket to choose Standard (S2) as Service Plan to maintain high performance and availability of the system.

Conclusion

The purposed Online UIA Booking Flight Ticket Web App is developed by using Microsoft Visual Studio 2017 with Visual Basic and ASP.net programming language. The Web App is published and hosted by Microsoft Azure cloud services, which enable the web app to handle more concurrent loads as well as maintaining high availability. Furthermore, the web app is hosted with 2 different endpoints provide failover. Whenever one of the endpoint fails, users will be redirected to an alternative endpoint on a different region, thus maintaining access the website without any restrictions.

In this assessment, developer able to adapt the valuable knowledge of developing web application using Visual Basic programming languages. Developer also discovered the usage of utilizing Azure Cloud service to benefit the Web App development. However, using learning and using Azure was a difficult task as its new to the developer. Nevertheless, the developer had gain a lot of knowledges while making mistakes and grateful to the guidance from lecturer as well as friend to complete this assignment.

References

1. Azure.microsoft.com. (2018). *Dev-Test deployment for testing PaaS solutions*. [online] Available at: https://azure.microsoft.com/en-us/solutions/architecture/dev-test-paas/ [Accessed 25 May 2018].

2. Community.oracle.com. (2018). *potty: Introduction to Class Diagrams Blog | Oracle Community*. [online] Available at: https://community.oracle.com/blogs/potty/2014/01/22/introduction-class-diagrams [Accessed 25 May 2018].

3. C-sharpcorner.com. (2018). *Connecting And Working With Azure SQL Database Using Visual Studio 2017*. [online] Available at: https://www.c-sharpcorner.com/article/connecting-and-working-with-azure-sql-database-usi/ [Accessed 25 May 2018].

4. C-sharpcorner.com. (2018). *Connecting And Working With Azure SQL Database Using Visual Studio 2017*. [online] Available at: https://www.c-sharpcorner.com/article/connecting-and-working-with-azure-sql-database-usi/ [Accessed 25 May 2018].

5. Docs.microsoft.com. (2018). *Connect to Azure SQL Data Warehouse - VSTS*. [online] Available at: https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-query-visual-studio [Accessed 25 May 2018].

6. Docs.microsoft.com. (2018). *Test your Azure web app performance under load from the Azure portal*. [online] Available at: https://docs.microsoft.com/en-us/vsts/load-test/app-service-web-app-performance-test [Accessed 25 May 2018].

7. Docs.rightscale.com. (2018). *Cloud Computing System Architecture Diagrams*. [online] Available at: http://docs.rightscale.com/cm/designers_guide/cm-cloud-computing-system-architecture-diagrams.html [Accessed 25 May 2018].

8. Techopedia.com. (2018). What is Load Testing? - Definition from Techopedia. [online] Available at: https://www.techopedia.com/definition/13649/load-testing [Accessed 25 May 2018].

9. Pal, K. (2018). *Understanding Volume, Load and Stress Testing*. [online] Mrbool.com. Available at: http://mrbool.com/understanding-volume-load-and-stress-testing/30019 [Accessed 25 May 2018].

10. plus (2018). *Types of Cloud Computing Explained | GlobalDots*. [online] GlobalDots - CDN, Security and Performance Solutions. Available at: http://www.globaldots.com/cloud-computing-types-of-cloud [Accessed 25 May 2018].