CT071-3.5-3-DDAC

Designing & Developing Cloud Applications

Individual Assignment

Ukraine International Airlines (UIA)

Online Flight Booking System

Name : Lam ZhanHui

TP Number : TP035832

Intake Code : UC3F1702IT(ISS)

Lecturer : Dr. Kalai Anand Ratnam

Date Due : 20th November 2017

Acknowledgement

Cloud application is new to the developer and it is a challenging task to master the concepts of cloud computing and develop cloud applications adhere to cloud design patterns in a packed final year schedule. The developer would like to express his deepest appreciation to the individuals who have provided comprehensive support throughout the period of designing and developing the cloud application.

First of all, the developer would like to express his genuine gratitude to the DDAC module lecturer, Dr. Kalai Anand Ratnam for his detailed lecture and results oriented approach which motivates us to complete the hands-on lab by practicing collaborative learning with Microsoft's resources such as Microsoft Teams allows us to gain and share knowledge with ease. Dr. Kalai has shared invaluable knowledge on the cloud computing and Microsoft Azure which benefits us in the perspective of cloud computing with the latest cloud design patterns.

Apart from that, the developer would like to convey his utmost gratitude to his family and friends who provide emotional support to developer. Their encouragement is among the pioneer motivation factor for the developer to complete the project on time.

Once again, the developer would like to thank all the individuals above that plays a crucial role to provide guidance and support to the developer throughout the journey of developing the Microsoft Azure cloud applications.

Table of Contents

1.0 Introduction

This project is focusing on developing a Flight Booking System web application for Ukraine International Airlines (UIA) with Microsoft Azure as the cloud platform. The Flight Booking System will be developed with ASP.NET and deploying on Azure. On the other hand, the web app is designed with the objectives of diverting the user traffic to specified endpoints with Azure Traffic Manager to ensure that users having seamless experiences while visiting the Flight Booking Web App. Apart from that, analytics of the web application will be ready for viewing and monitoring the performance of the web application. The flight booking web app is expected to have the scalability in order to withstand the high demands throughout peak seasons which is among the objectives of this project, being scalable dynamically based on the traffic.

1.1 Objectives

The objectives of the project include:

- To design and develop an Online Flight Booking System for Ukraine International Airlines (UIA)
- To deploy the Online Flight Booking System on Microsoft Azure as an App Service
- To analyze the performance of the web application with the tools available on Microsoft Azure
- To scale the web application according to the demands such as peak seasons
- To configure an unremitting deployment workflow for the UIA Online Flight Booking System on Azure with the use of GitHub
- To conduct performance testing on the web application based on various user load and analyze the collected data

1.2 Scope

The scope of the project is to design and develop an Online Flight Booking System for Ukraine International Airlines (UIA) and deploy it to Microsoft Azure as a single tenant App Service. The web application will allow the customers to manage the whole flight book process which the customers can create an account prior booking the flight tickets. On the other hand, the app service will have a traffic manager profile which enhance the overall performance of the web application. Apart from that, performance testing will be conducted to analyze the scaling of the web application.

1.3 Requirements

1.3.1 Functional Requirements

- Visitors can view the flight schedule on the webpage
- Visitors can search for available trip on the webpage
- Visitors of the webpage have an option to register as a member
- Registered users can book the flight tickets
- Users can view the flight details after the booking has been made
- The flight booking system have a database to store the user details and flight booking details

1.3.2 Non-functional Requirements

- Capable to maintain the response time below 100ms with 500 concurrent users on the webpage
- Scalable to cover the demand on peak seasons with high traffic
- Able to route the users to endpoints in different geographical locations to achieve responsive page load time

1.4 Summary of major functions

Ukraine International Airlines (UIA) Online Flight Booking system is a web application powered by Microsoft Azure where it is designed for external usage which is the flight customer. As it is a single tenant web application, Azure Active Directory will not be utilized in the UIA Online Flight Booking System.

The system allows the guests to view the trip served by UIA and register as a customer to manage the flight booking processes from selection of trip to checkout.

2.0 Project Plan

2.1 Gantt Chart

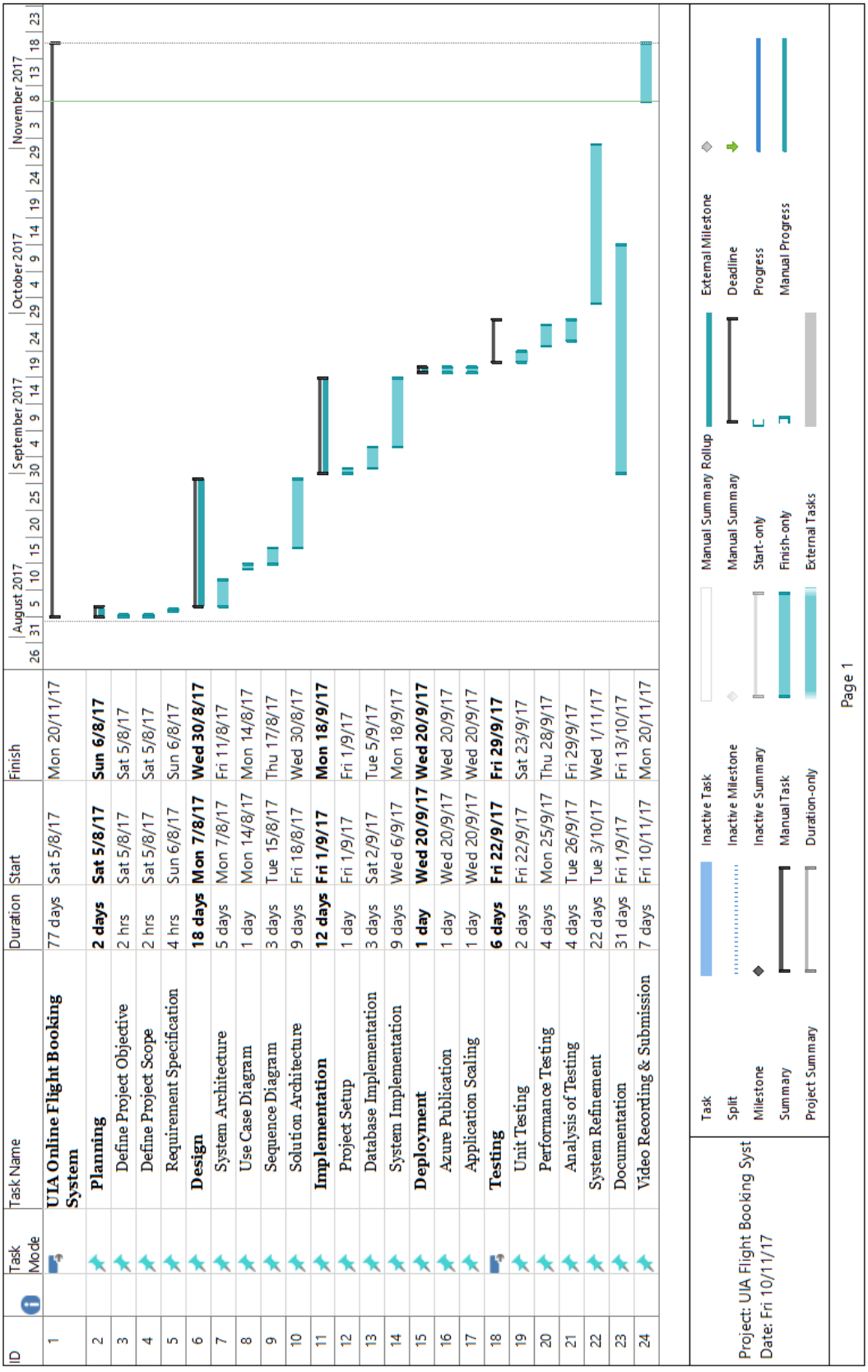


Figure 2.1 Gantt Chart of the Project

3.0 Design

3.1 Cloud Design Patterns

According to Cecaro (2015), there are 24 cloud design patterns in Azure which is categorized into 8 Problem Areas as displayed in the figure below.

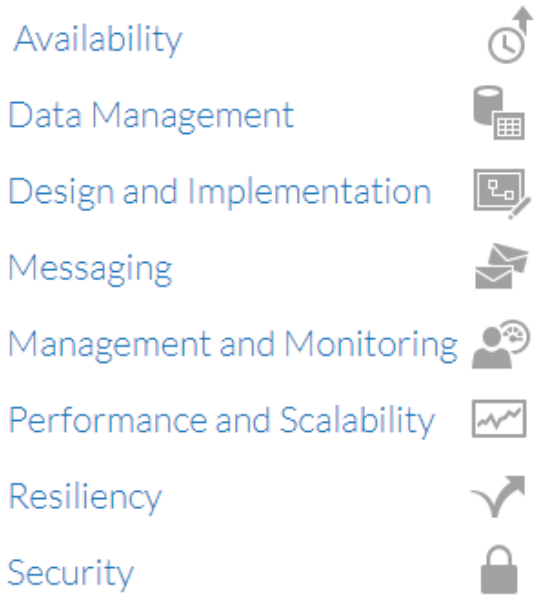


Figure 3.1 8 Problem Areas which the Cloud Design Patterns categorized into (Cecaro, 2015)

In the UIA Online Flight Booking App Service, some of the design guidance and patterns are utilized to solve the problem areas.

Availability – SQL Server database Geo Replication has been implemented to ensure the availability of the UIA cloud application where the databases will be replicated on a different region. Also, the azure storage that will store the audit logs of the database will be replicated in three datacenters by default (Microsoft, 2017).

Management and Monitoring – Health Endpoint Monitoring Pattern is adhered where the built-in monitoring features of Azure has been utilized to monitoring the endpoints of the application published on Azure. Alert rules can be created on the application such as the websites to raise the alert when the system is malfunction.

Performance and Scalability - Auto scaling is used to allocate the resources dynamically to equal the performance of requirements defined. To illustrate, in the Web Application of UIA Online Flight Booking System has been configured to scale-out automatically when CPU and memory resources have hit the pre-defined values.

3.2 Architectural Diagrams

3.2.1 Use Case Diagram

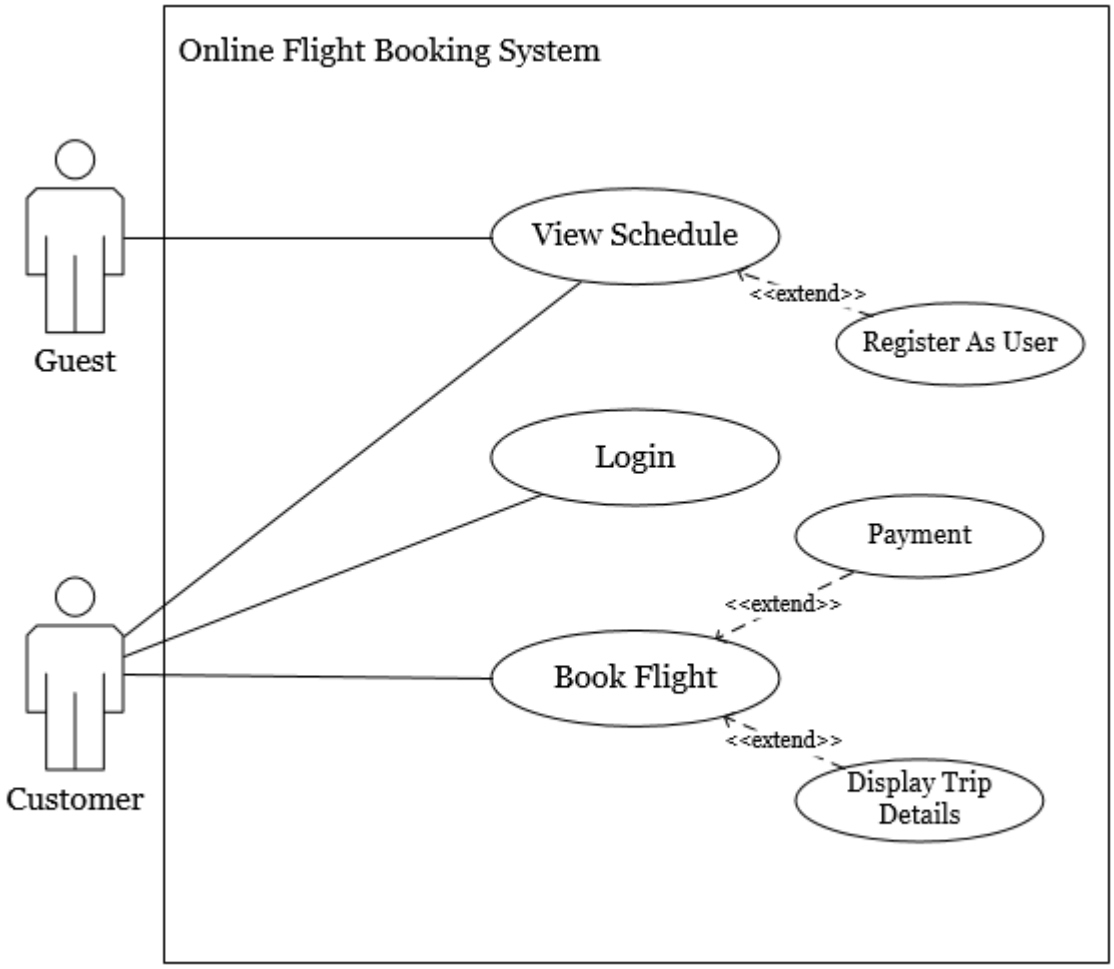


Diagram 3.2.1 Use Case Diagram of Ukraine International Airlines (UIA) Online Flight Booking System

The use case diagram of the UIA Online Flight Booking System involves two primary users which are guest and customer. Guest have the function of viewing the schedule only where they have to register in order to perform the booking. The process is to encourage the automation process when the users booking the flight with an account, the user details will be entered the flight booking details. Therefore, the users are required to enter their details such as name, gender, passport number during the registration only and the details will be populated during the flight booking process.

3.2.2 Use Case Specifications

Use Case	View Schedule
Actor	Guest, Customer
Extend	Register as user
Pre-conditions	Visitors have to access to the website to view the schedule
Post-conditions	Flight schedule will be displayed to the user
Flow of Events	1. Enter the website url 2. Navigate to flight schedule page 3. Apply destination search filter 4. Flight schedule listed according to filter applied
Alternative Path	4.1 Register as a user a. Enter Name and other essential details b. User account created

Table 3.2.2.1 Use Case Specification for View Schedule

Use Case	Login
Actor	Customer
Pre-conditions	Registered with an account before logging in into UIA Online Flight Booking Systems
Post-conditions	Web Page with user details loaded
Flow of Events	1. Enter the website url 2. Login with user credentials 3. Webpage with user details loaded
Alternative Path	2.1 Incorrect Login Credentials entered a. Reenter Login Credentials

Table 3.2.2.2 Use Case Specification for Login

Use Case	Book Flight
Actor	Customer
Extend	Payment, Display Trip Details
Pre-conditions	To book the flight, webpage visitors have to register as a user first
Post-conditions	Flight Details displayed after payment has been made
Flow of Events	1. Enter the website url 2. Login with user credentials 3. Navigate to flight schedule page 4. Select desired trip 5. Enter booking details 6. Confirm booking details 7. Proceed to payment 8. Booking confirmation details returned
Alternative Path	2.2 Incorrect Login Credentials entered a. Reenter Login Credentials 5.1 Selected trip has insufficient seats a. To select other similar trips

Table 3.2.2.3 Use Case Specification for Book Flight

3.2.3 Sequence Diagram

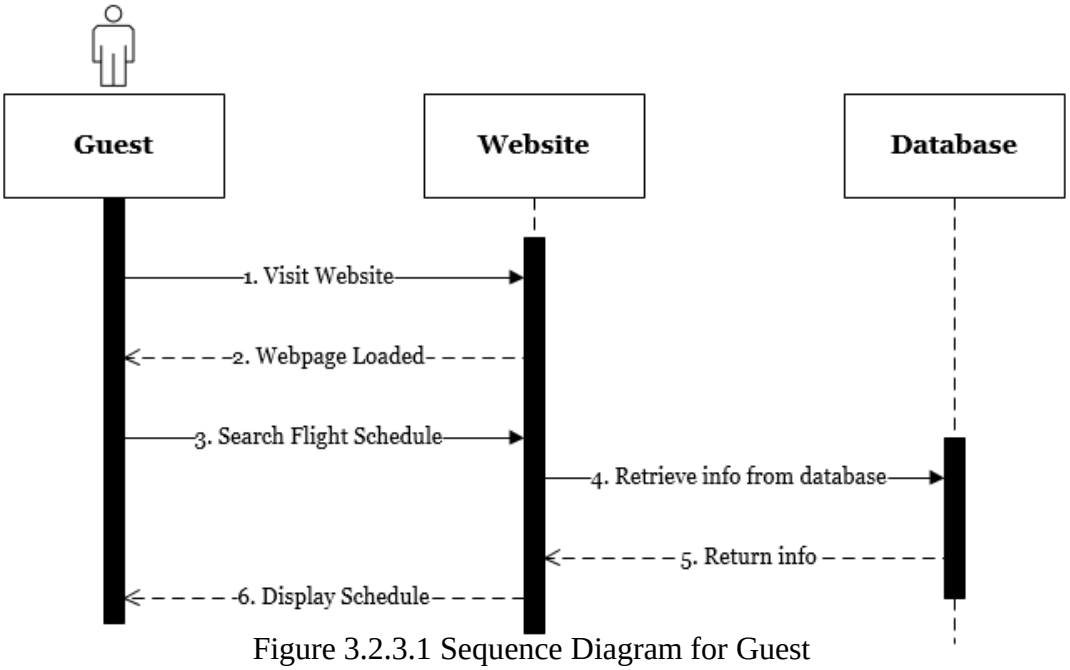


Figure 3.2.3.1 Sequence Diagram for Guest

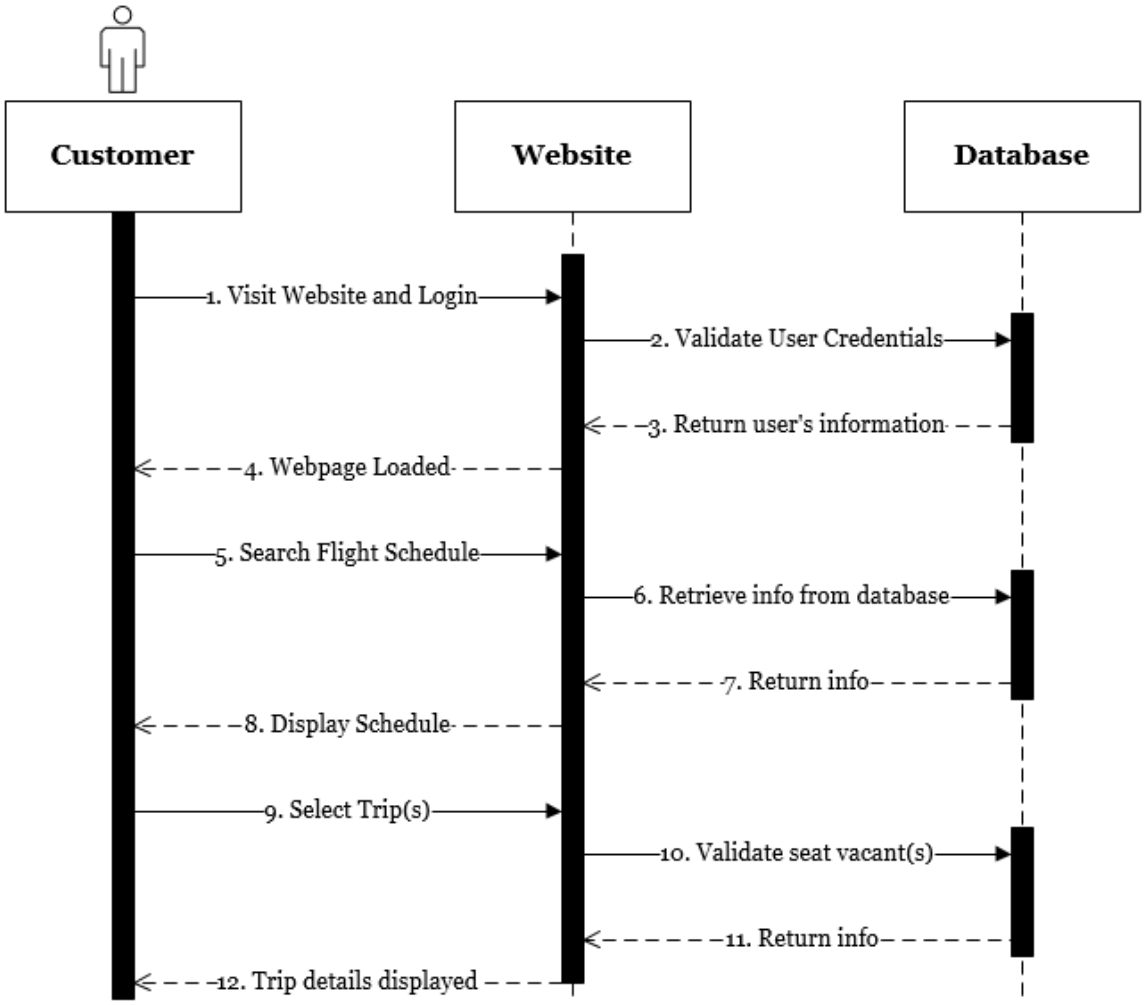


Figure 3.2.3.2 Sequence Diagram for Customer

3.3 Design Considerations

The considerations while designing the Online Flight Booking System includes the requirements defined by the authorities of UIA and the assumptions on the budget to utilize the resources available on Azure. To illustrate, UIA have the goals of expanding the operation beyond Ukraine is handicapped by the scale of the website where the performance is subpar beyond Ukraine. Therefore, the developer has taken the scalability of the web application into the consideration and ensure that the website is scalable during the peak seasons to ensure that the UIA Online Flight Booking Website has an optimum performance. Also, with a good scalability of the website ensures that it leaves a good impression to the customers, especially on the recently expansion area such as South East Asia.

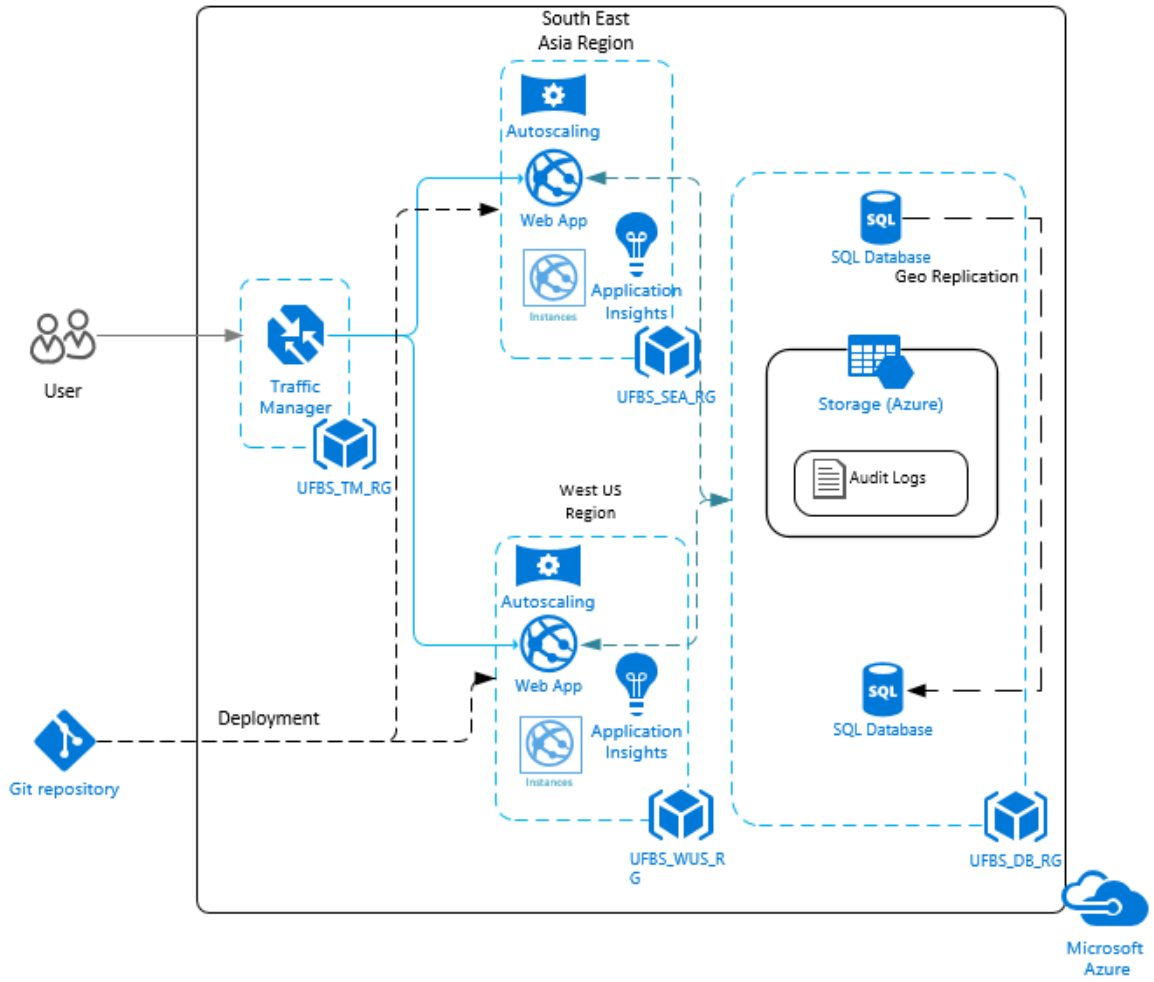


Figure 3.3 Cloud Architecture Diagram of UIA Online Flight Booking System

In the perspective of the Azure resources, the utilization of the services offered by the Azure is based on the budget of RM150 per month. Therefore, the auto-scaling instances and the

database tier selected for the UIA Online Flight Booking System will be spent according to the budget. Apart from that, performance testing will be done to ensure that the app service developed is able to meet the demands of the users from various geographical locations.

The UIA web application is deployed on Azure with 2 app services where one of the app service is on West US 2 region whereas South East Asia (SEA) is selected for another app service. The endpoints of the 2 app services are handled by traffic manager to ensure users of the web app will experience the best performance. Also, the availability of the system is considered by having an active geo-replication to keep the web app operational when the primary database is down.

Application Insights has been configured to monitor the performance of the web app and be able to detect performance issues with the live system. Also, the source code will be placed in GitHub, the selected source control management services.

4.0 Implementation

The list of Azure services applied in the Ukraine International Airlines Flight Booking System are illustrated in the figure below.

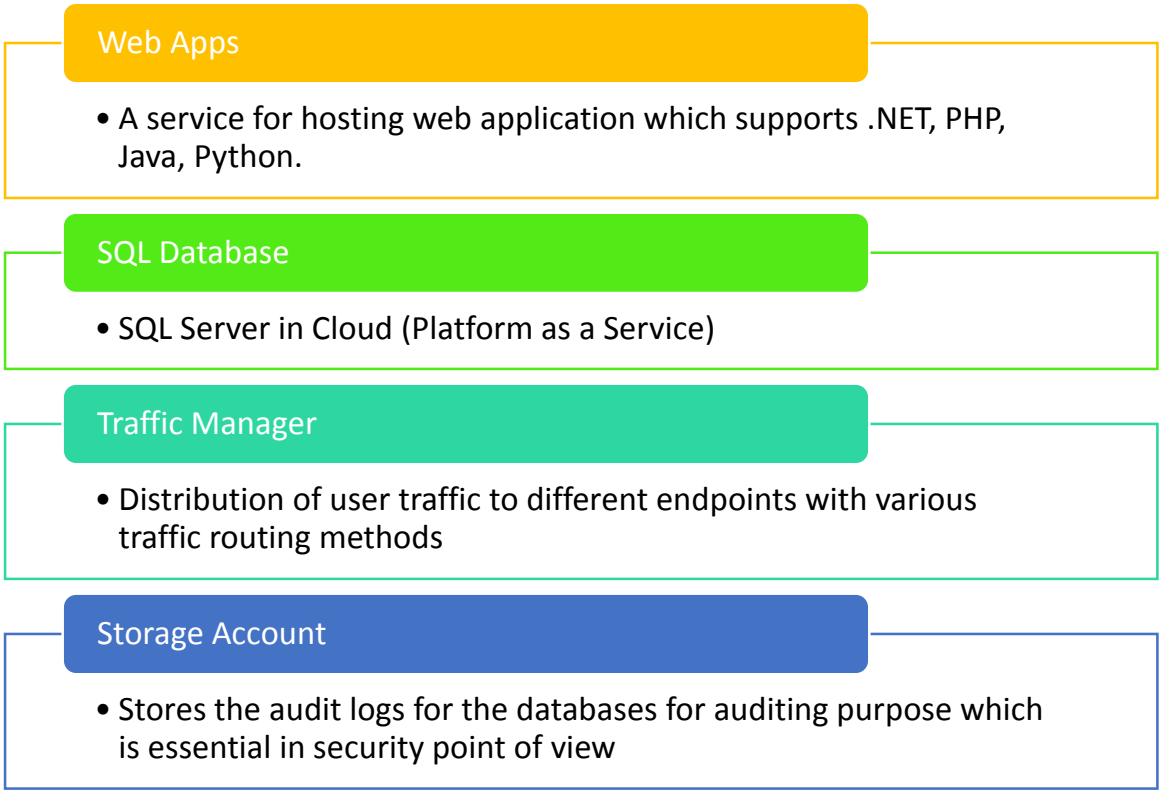


Figure 4.0 Azure Services utilized for UIA Online Flight Booking System Web App

4.1 Web App publication to Azure

4.1.1 Create Resource Group

Prior to the publication of the UIA Flight Booking System on Microsoft Azure platform, resource groups will be created. Resource group is a container that allows the developer to hold related resources which allows to resources to be deployed, managed and monitored as a group rather than individually (Macken, et al., 2017).

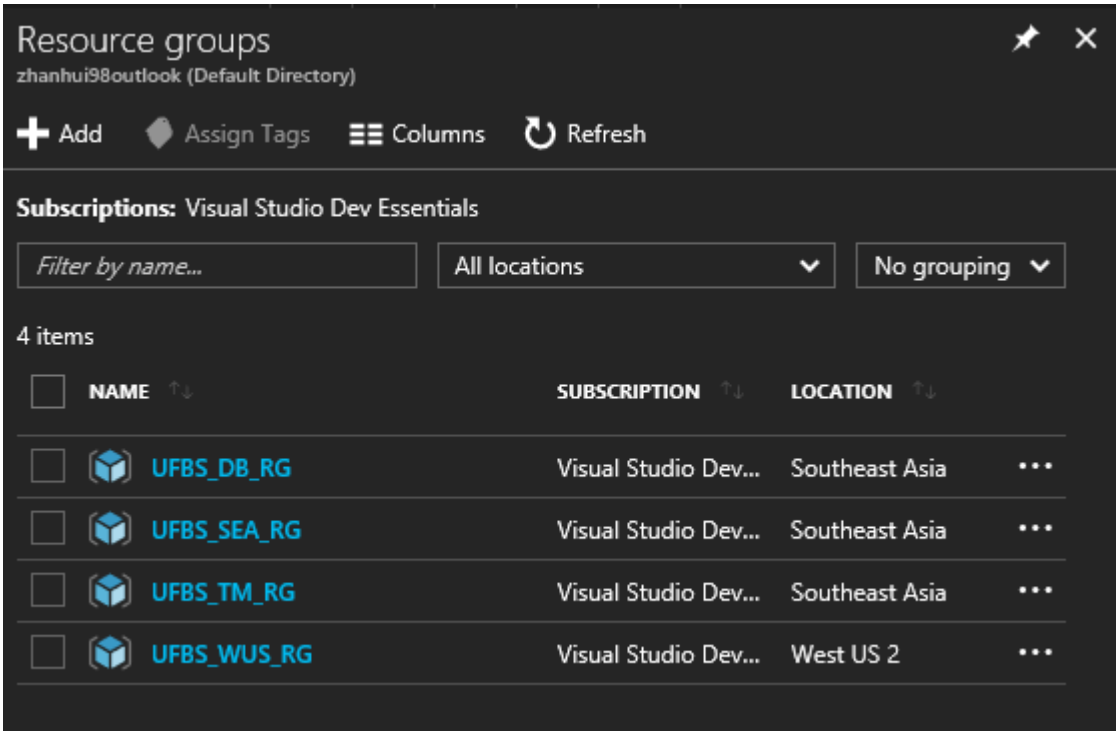


Figure 4.1 Resource Group for the UIA Flight Booking Web App

There are 4 resource groups created for the UIA Flight Booking Web App.

Name of Resource Group	Description
UFBS_SEA_RG	App service in South East Asia region
UFBS_WUS_RG	App service in West US 2
UFBS_DB_RG	Storage account and SQL Database
UFBS_TM_RG	Traffic Manager

Table 4.1.1 Resource Groups for UIA Flight Booking Web App

Each resource group has been given a distinctive name to differentiate its purpose. To illustrate, the UFBS_SEA_RG is created to store the app service of South East Asia region whereas the UFBS_WUS_RG is used to store the app service on the different region, which is West US 2. South East Asia region is selected to deploy the app service as the Ukraine International Airline

targets to expand its operation in the region of South East Asia. UFBS_DB_RG and UFBS_TM_RG are resource groups for storage account, SQL Database and traffic manager profile respectively.

4.1.2 App Service

Lin (2017) advocates that Azure Web Apps is a web application hosting service which allows load balancing, autoscaling, security and automated management. Apart from that, the Web Apps does offers DevOps capabilities which includes continuous deployment from GitHub and Visual Studio Team Services (Lin, 2017).

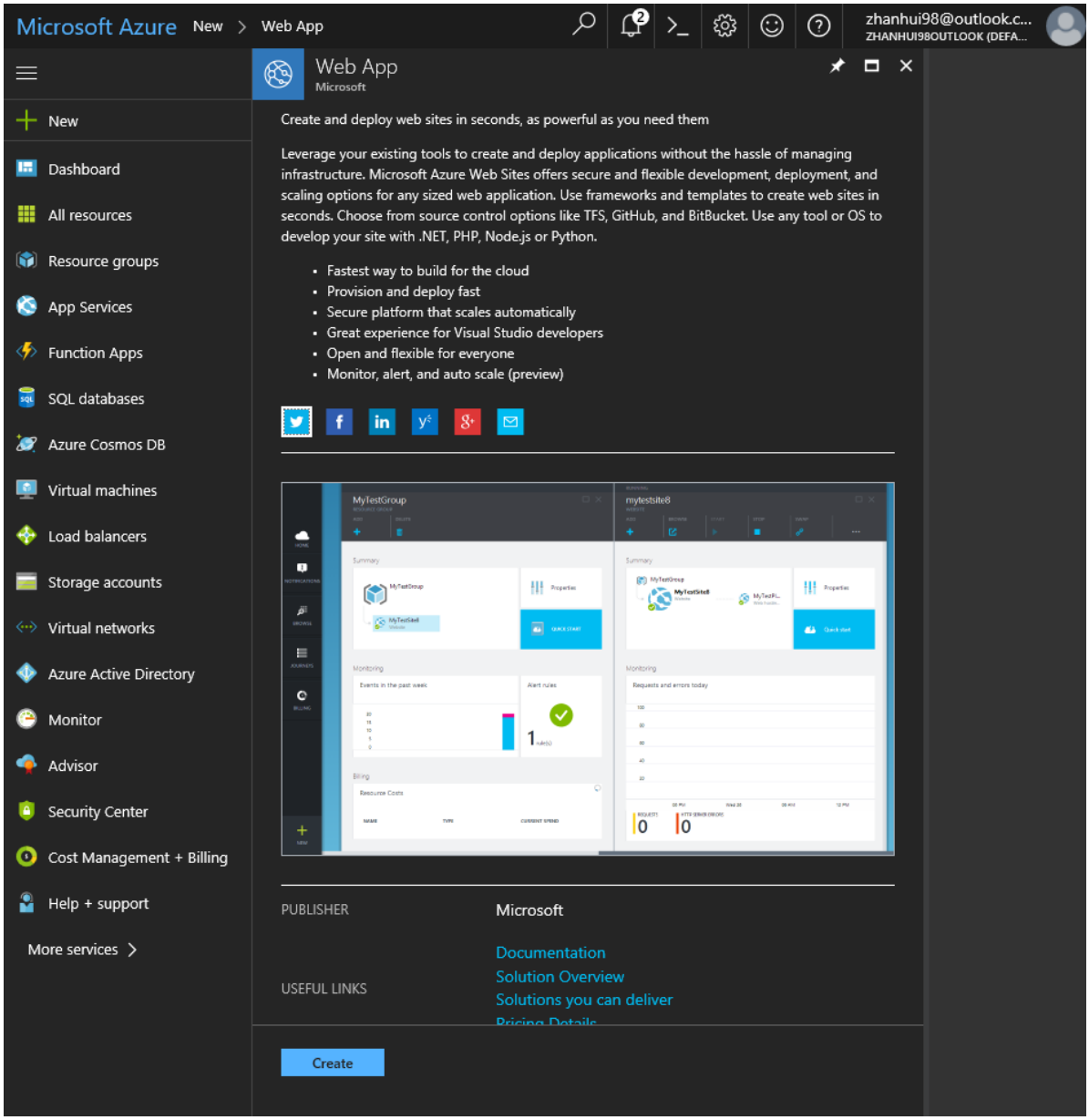


Figure 4.1.2 Creation of App Service in Azure

The app service for the UIA Flight Booking System will be created for South East Asia Region and West US 2.

Configurations of Web App

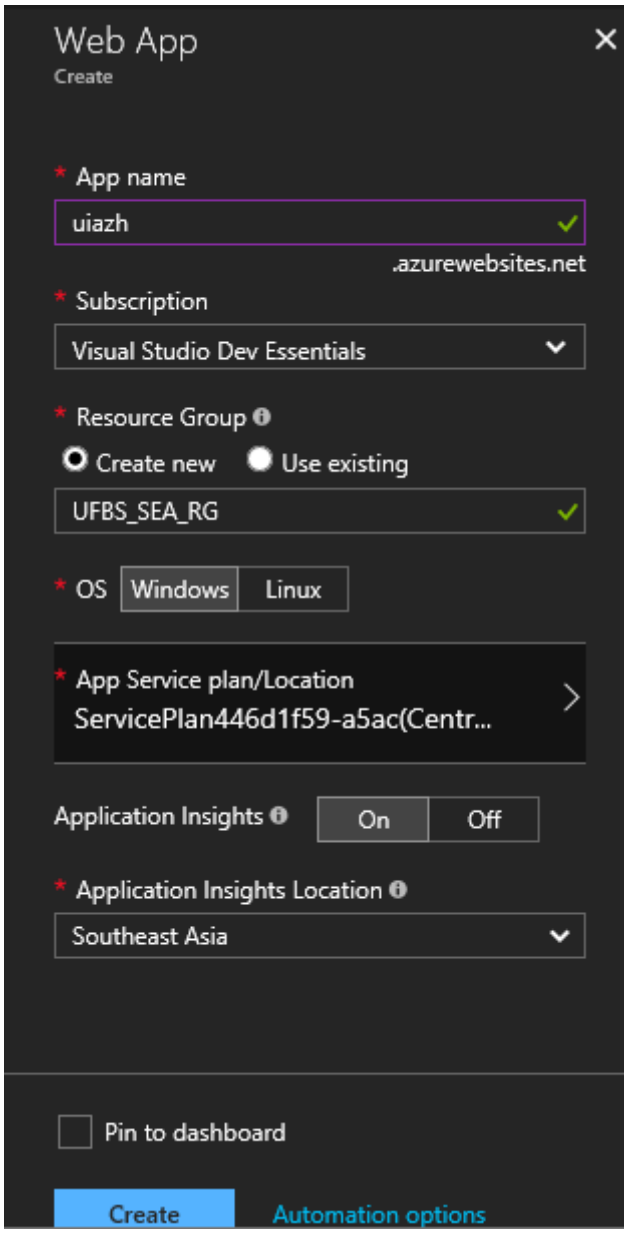


Figure 4.1.2.1 Configuration of App Service

The web application has been given the a unique name of "uiazh" where the URL of the website will be displayed as (uiazh.azurewebsites.net). The web app will be placed in the resource group of UFBS_SEA_RG. The Operating System selected is Windows whereas the App Service plan will be configured. Furthermore, Application Insights has been turned on for performance monitoring.

Create a new service plan

A new App Service Plan is created for the UIA Online Flight Booking System to tailor with the business requirements of the Ukraine International Airlines.

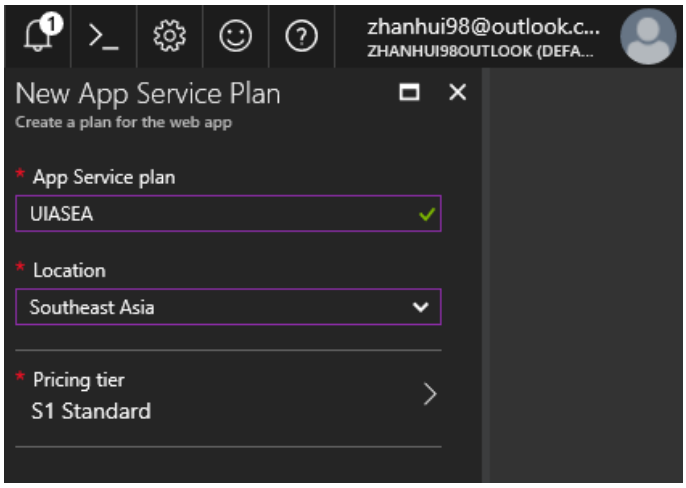


Figure 4.1.2.2 App Service Plan

The App Service Plan has been given the name of "UIASEA" for the app service in South East Asia in order for the team to differentiate the region of the web app. The location is set to Southeast Asia with the pricing tier of S1 Standard.

There a total of 6 different pricing tier of the App Service Plan which includes:

- Free
- Shared
- Basic
- Standard
- Premium
- Isolated

Among the pricing tier, there are different range such as S1, S2, S3 in Standard and B1, B2, B3 in Basic. Some of the pricing tier are displayed in Figure 4.2.2.3.

These are some of the pricing tier available on the Azure Web App.

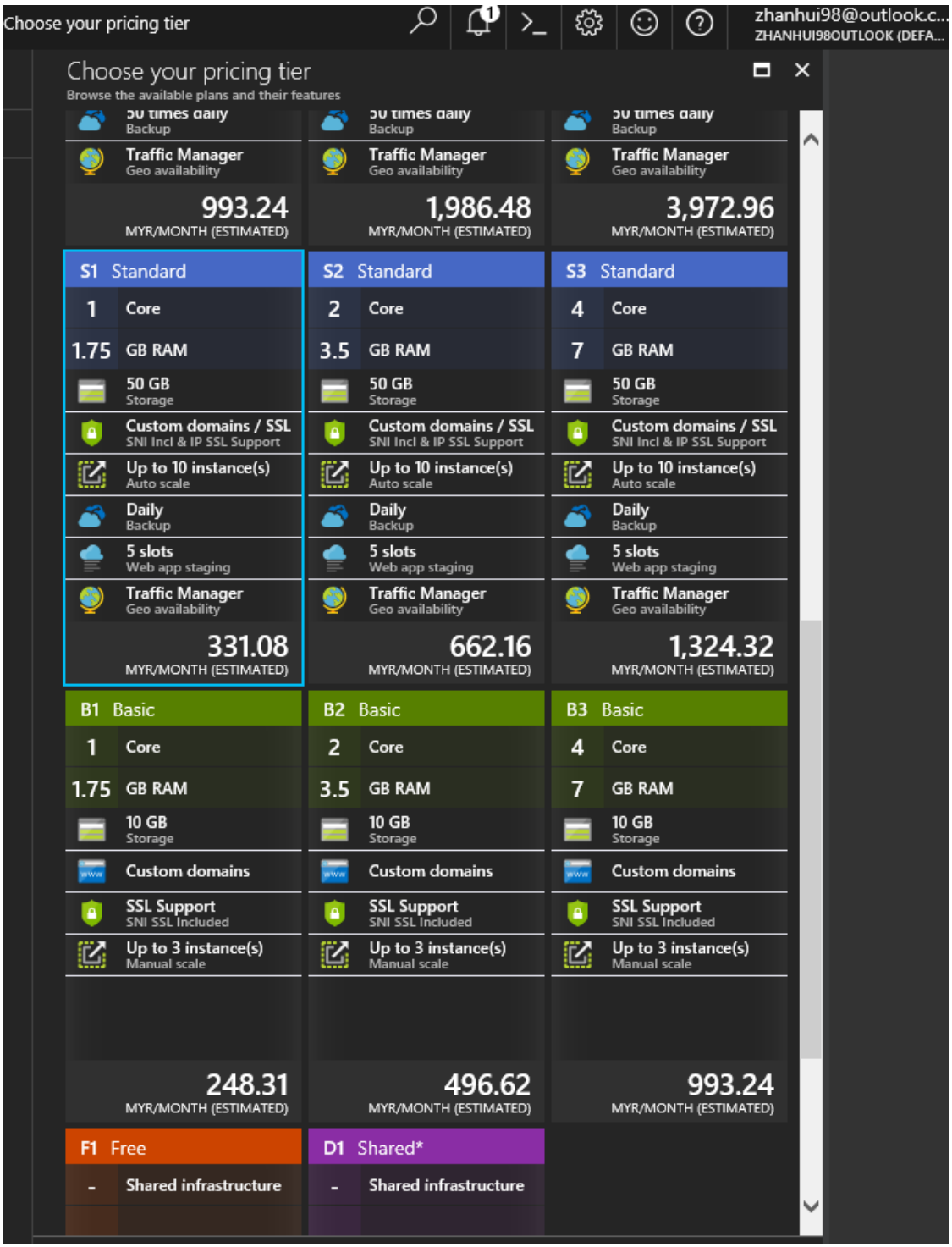


Figure 4.2.2.3 Pricing Tier Available for the App Service Plan

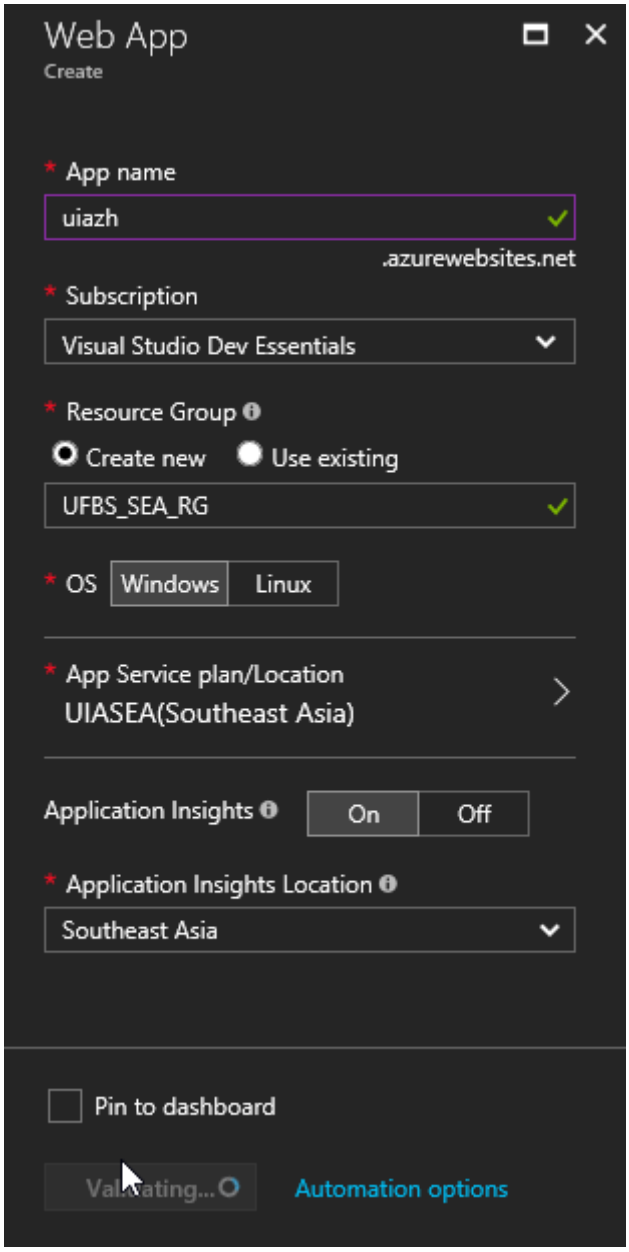


Figure 4.2.2.4 Final Configuration of the App Service

The figure 4.2.2.4 shows the final configuration of the Web App where the App Service plan and the location has been configured to UIASEA in Southeast Asia.

A similar app service is created for West US 2 Region with a new app service plan of UIAWUS with the location of West US 2.

Discussions on the selected plans

S1 Standard has been selected for both App Service in Southeast Asia and West US 2 of the UIA Online Flight Booking System. Justification of the selection can be found at section 4.2.

4.1.3 Azure SQL Database with Geo-Replication

SQL Server is vital to host the SQL database. The SQL Server will be created on the UFBS_STORAGE resource group along with the SQL Database. There will are two SQL Servers and SQL Databases will be created in the resource group where Southeast Asia will be the primary and West US 2 is the secondary database. The walkthrough will cover the creation of SQL database and SQL server on the Southeast Asia region.

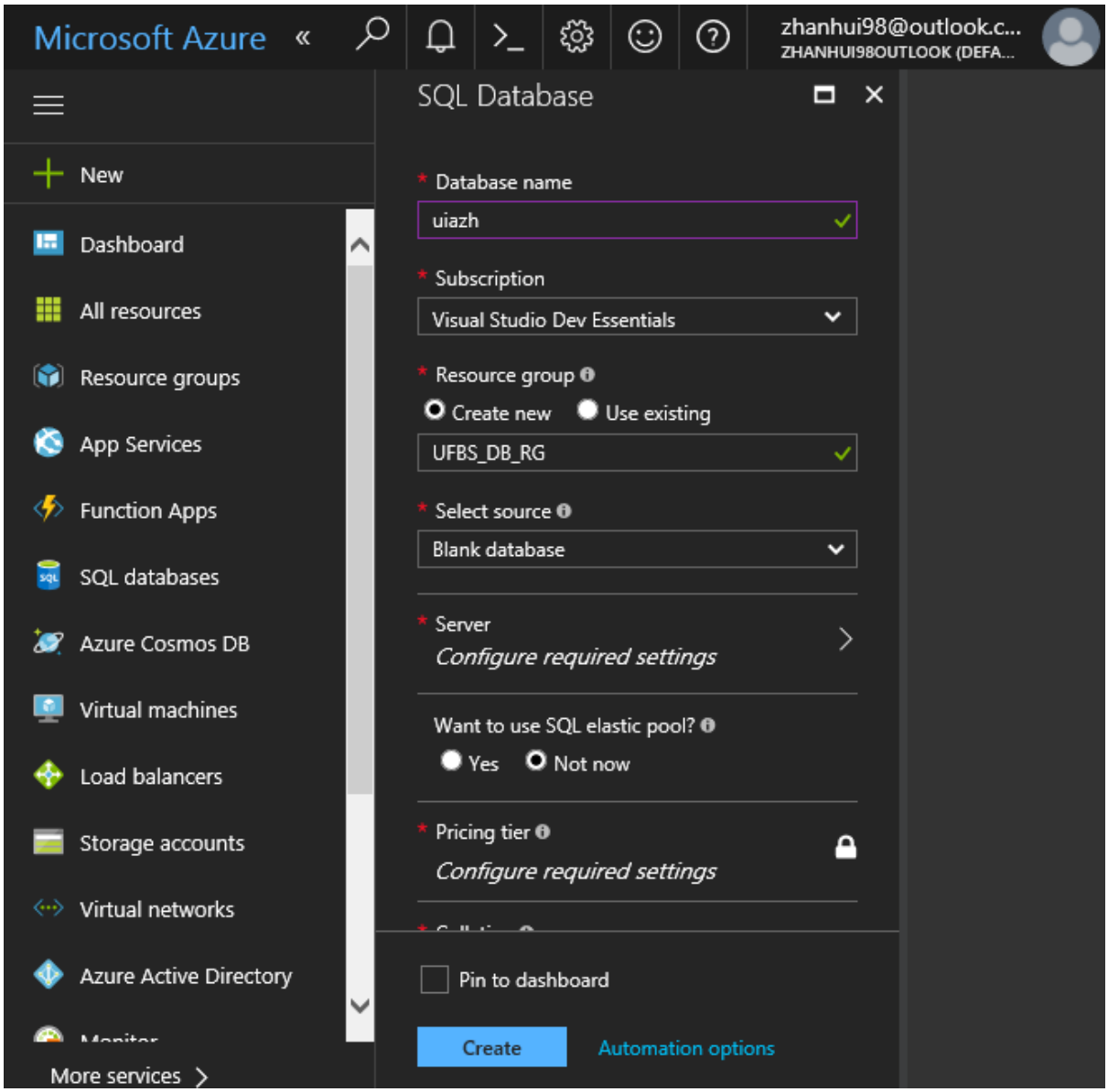


Figure 4.1.3 Creation of SQL Database

Configure SQL Server

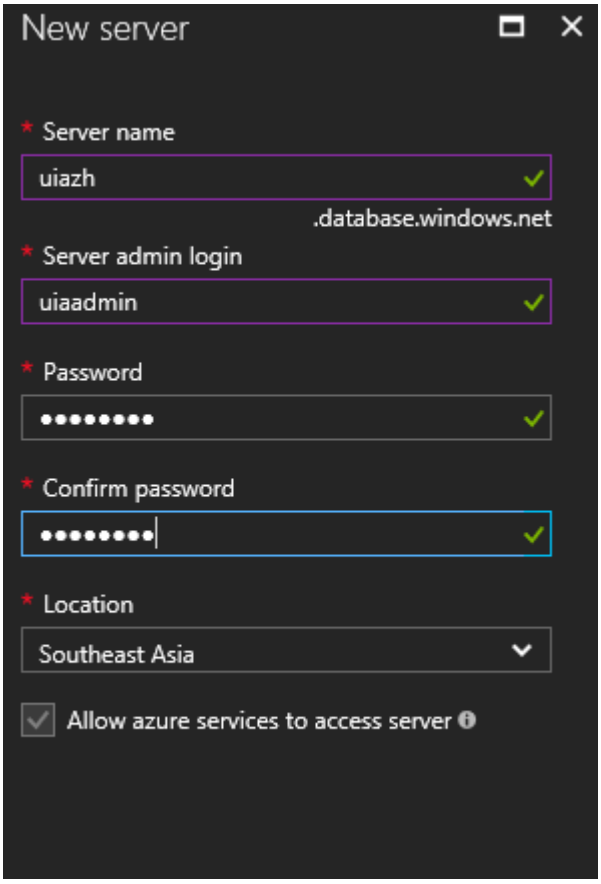


Figure 4.1.3.1 Configuration of SQL Server

The SQL Server has been created with the login account of "uiaadmin".

Selection of Pricing Tier

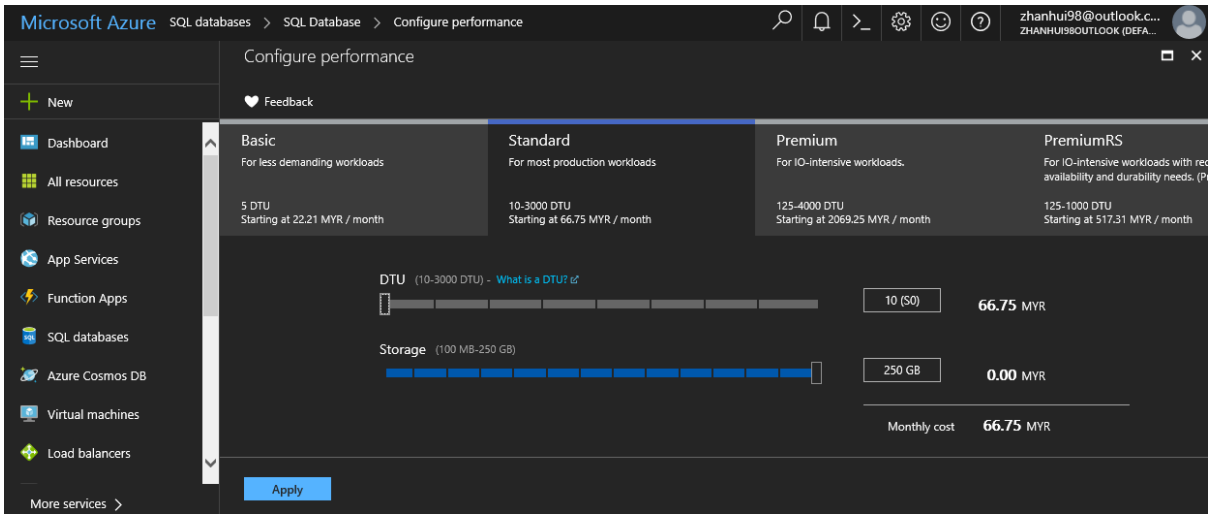


Figure 4.1.3.2 Pricing Tier of SQL Database

A DTU is a data throughput unit where it serves as a relative measure of the available power behind the database (Allen, 2017). DTU is a unified measure of CPU, memory, reads and writes on the performance (Rabeler & Kerkhove, 2017). The selected pricing tier for the UIA Online Flight Booking System is **Standard S0** where 10 DTU and a maximum of 250GB Storage is available. The justifications for the pricing tier of the database can be found at section 4.2.

These are configuration of the SQL Database on Azure prior creating it.

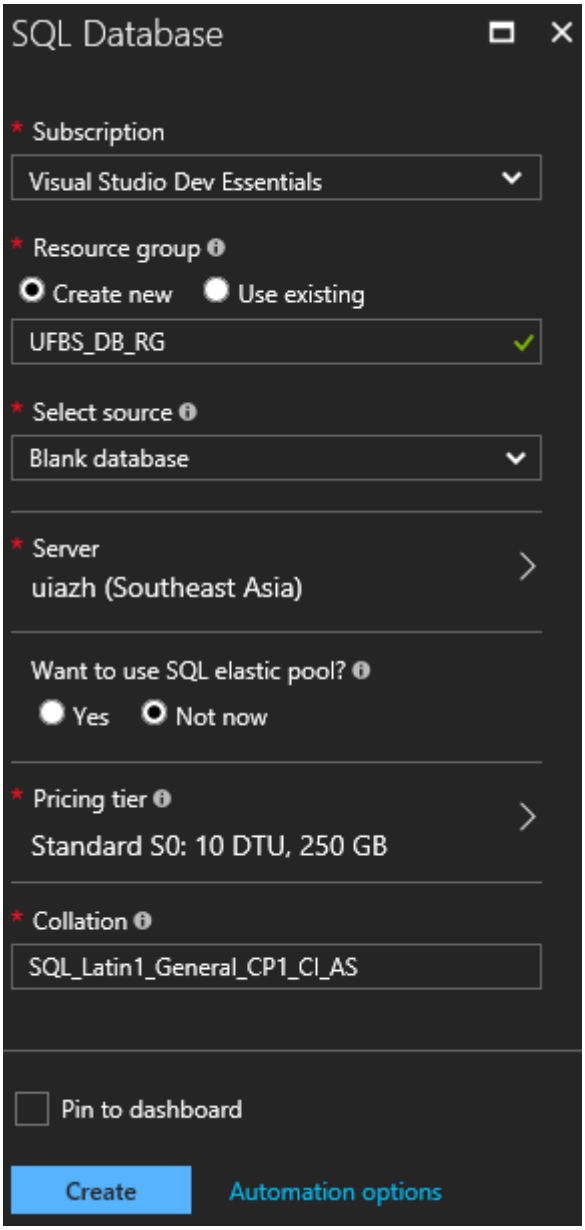


Figure 4.1.3.3 Configuration of SQL Database for Web App hosted in Southeast Asia

Utilizing Features Available on Azure

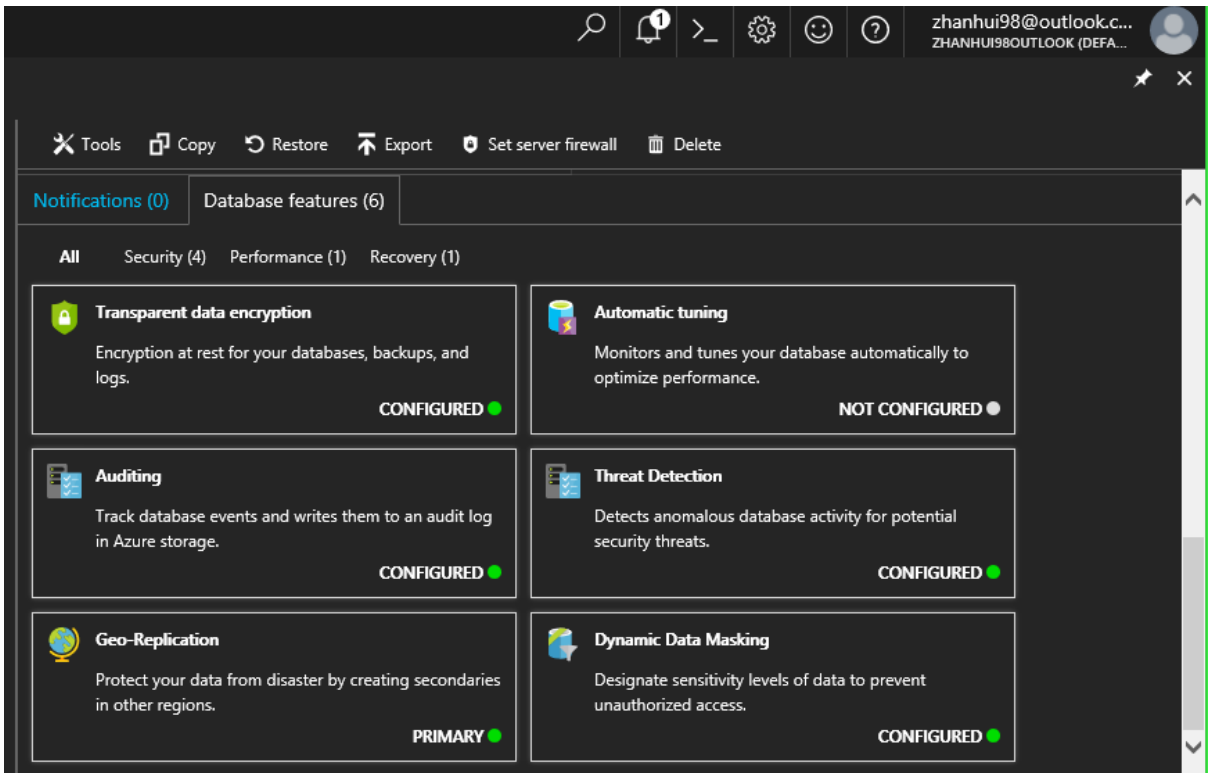


Figure 4.1.3.3a Features on Azure SQL Databases

The features such as Dynamic Data Masking has been configured to avoid unauthorized access on the data and Auditing Threat Detection to detect suspicious database activities.

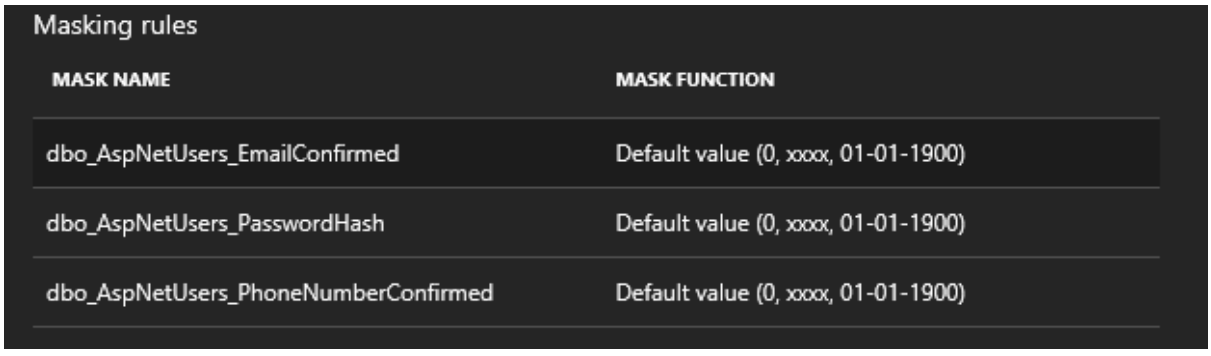


Figure 4.1.3.3b Masking Rules on the UIA Flight Database.

Figure 4.1.3.3b shows the masking rules on the database which includes the email, password hashed and phone number of the customers.

Create secondary database for Geo-Replication

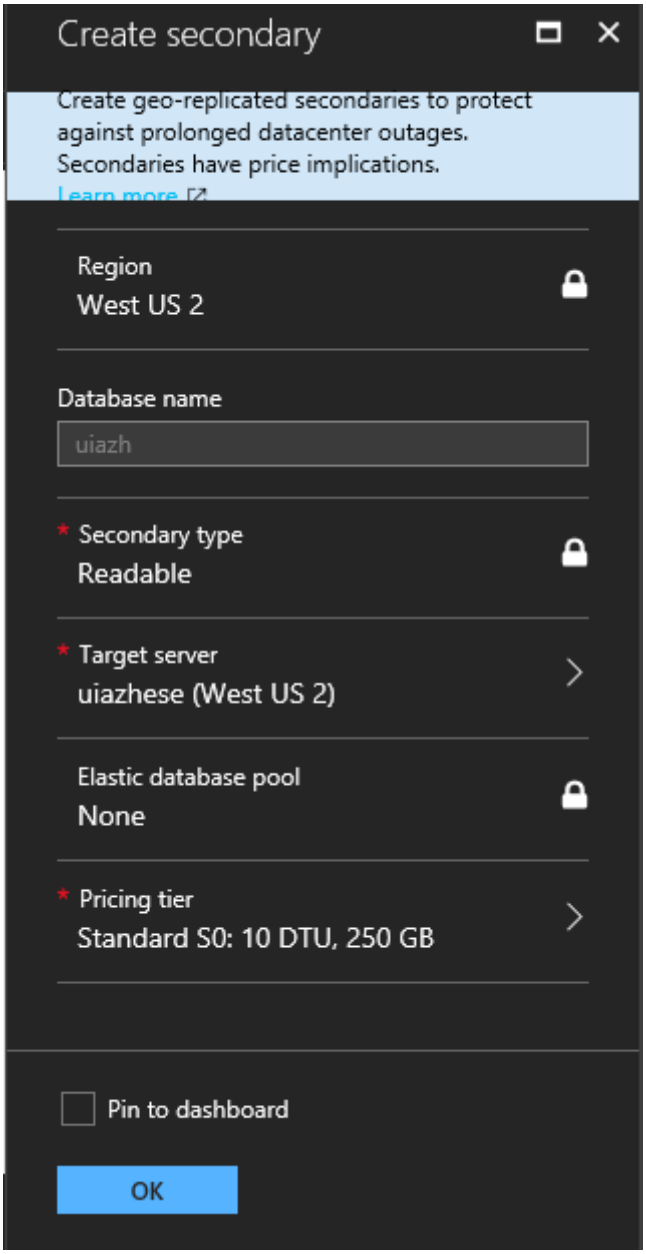


Figure 4.1.3.4 Configuration of Secondary SQL Database (West US 2)

West US 2 is the region for the secondary database and the type of the database is **Readable**. The pricing is similar to the one in the primary database which is Standard S0. Allen (2017) suggested that the pricing tier of the primary and secondary database has to been the same to avoid the performance related issues when different database is accessed by the web application.

The Database Replication process can be seen in the dotted line.

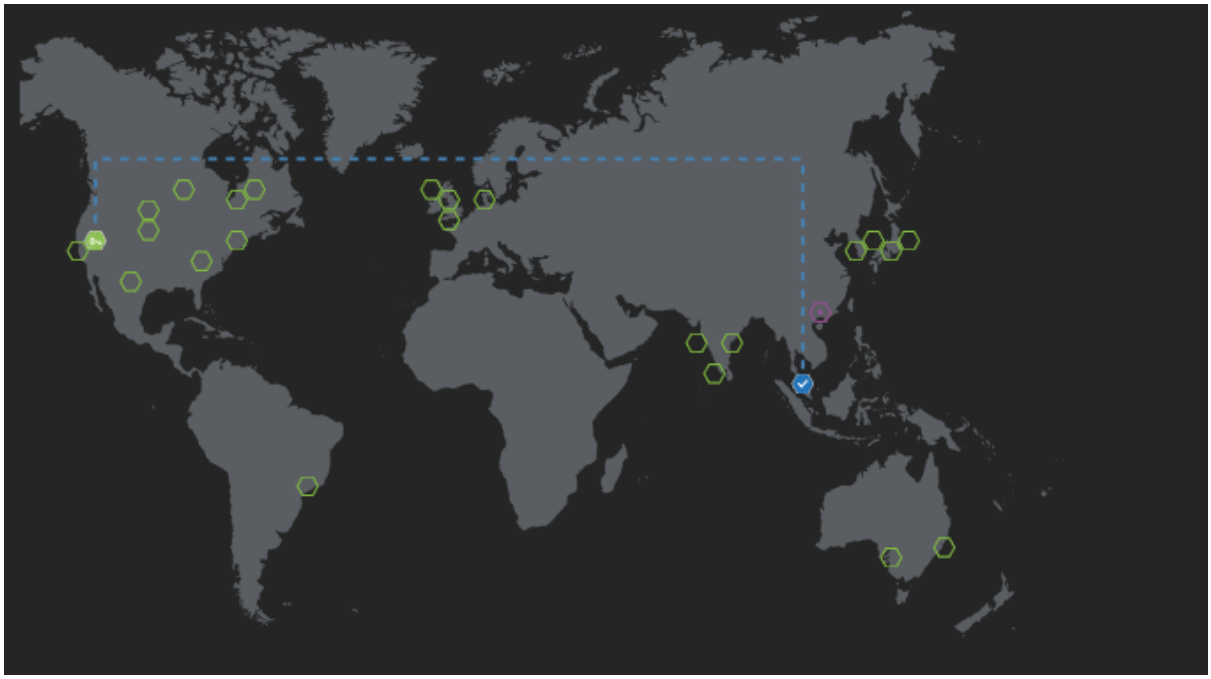


Figure 4.1.3.5 Data Replication Process to Secondary SQL Database (West US 2)

The dotted line turns into a solid line when the replication process is completed.

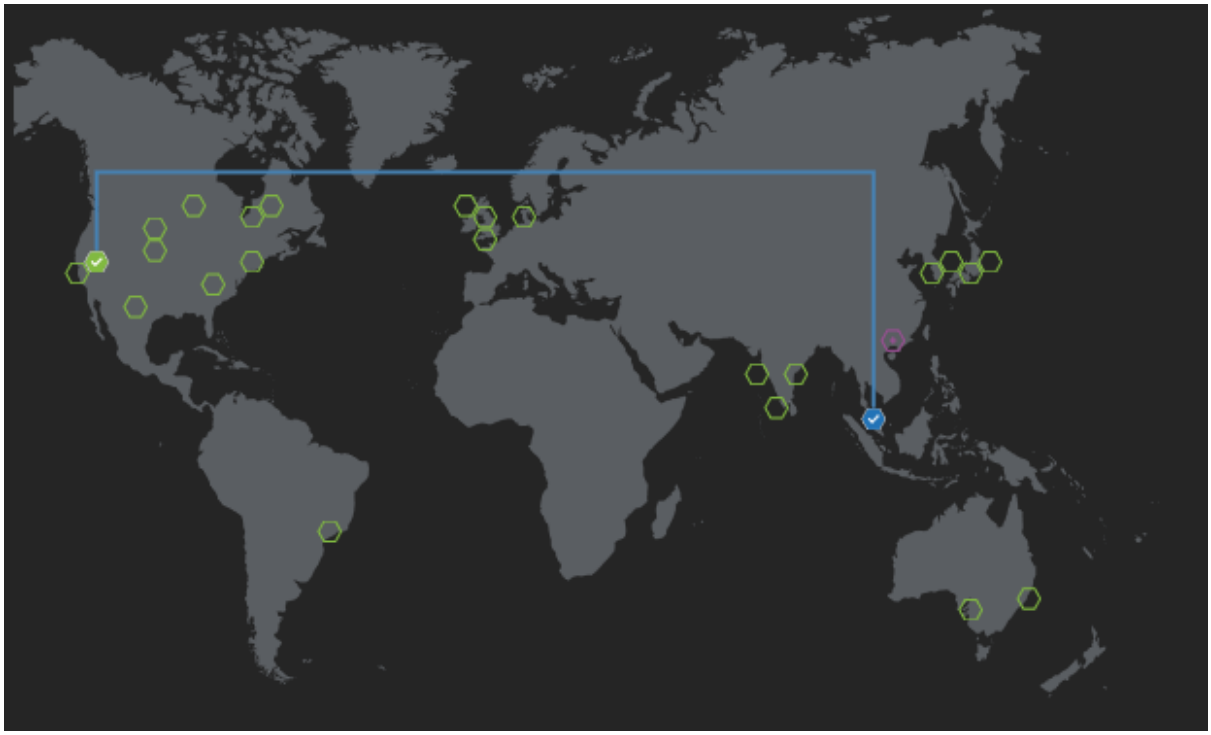


Figure 4.1.3.6 Completed replication to Secondary SQL Database (West US 2)

The Geo-Replication of the database will be displayed as below. Failover policy to be configured later.

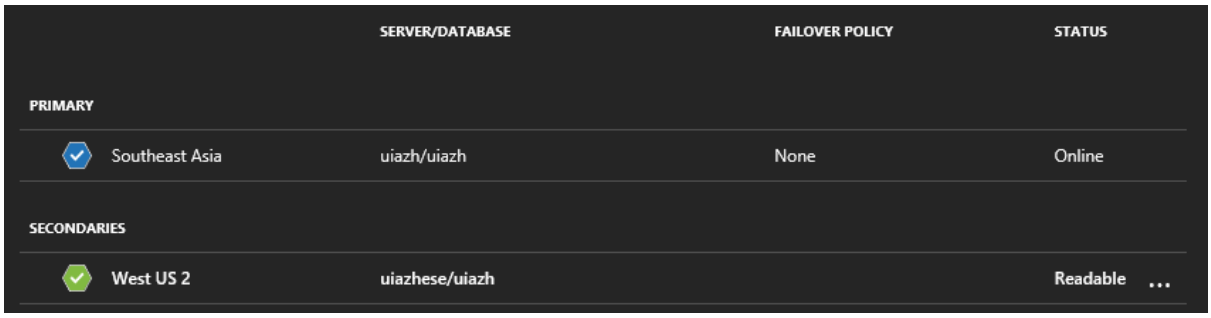


Figure 4.1.3.7 SQL Database Geo-Replication

Configuration of Failover policy

Failover policy is created through the primary database's Failover Groups.

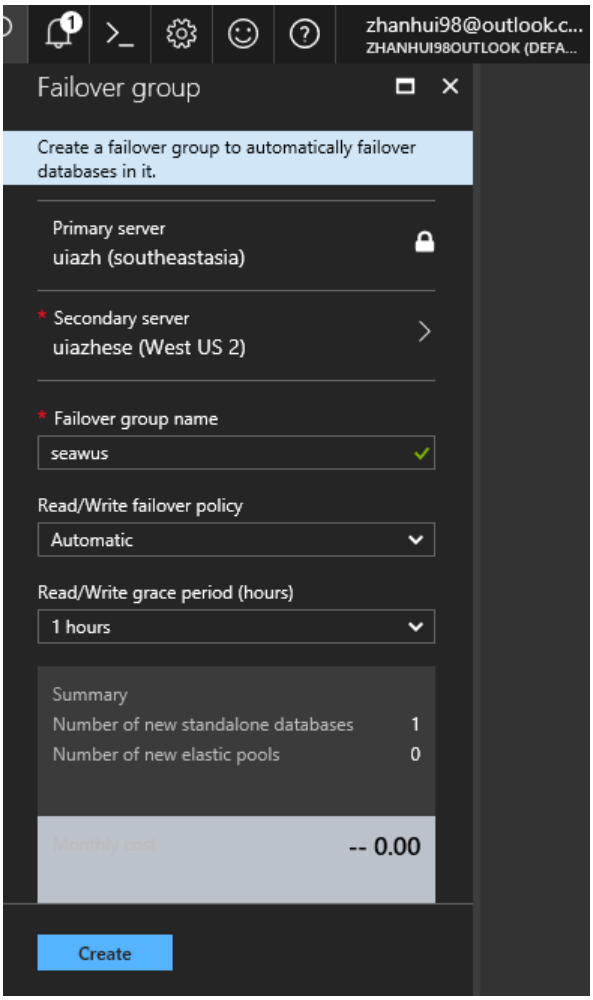


Figure 4.1.3.8 Configuration of Failover Policy on Primary Database

The failover group has been given the name of seawus with the Read/Write policy is configured to automatic with 1 hour of grace period.

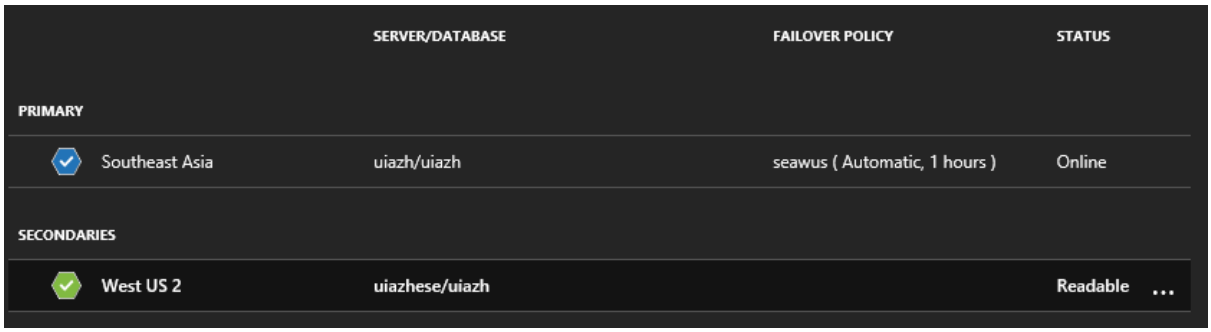


Figure 4.1.3.9 SQL Database Geo-Replication with failover policy configured

4.1.4 Publication of ASP.NET UIA Online Flight Booking System Web App to Azure

In Visual Studio, the web app will be connected to the SQL Database created in Azure portal.

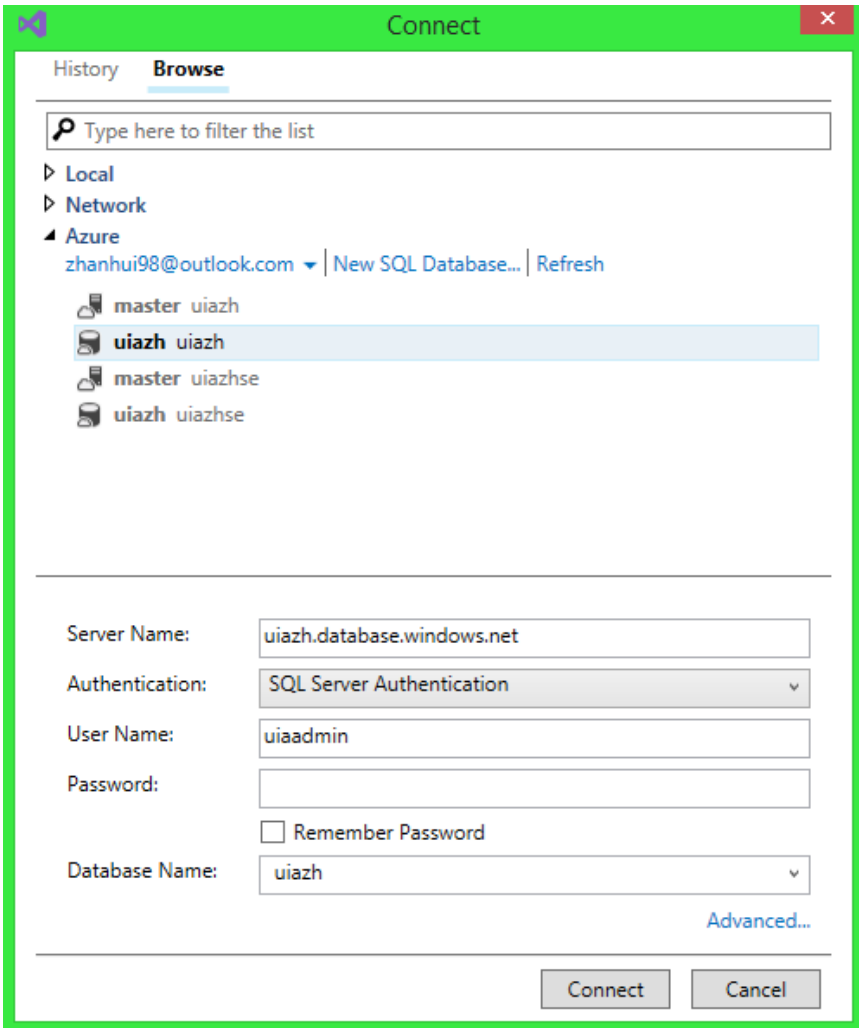


Figure 4.1.4.1 Connect local app to Azure SQL Database

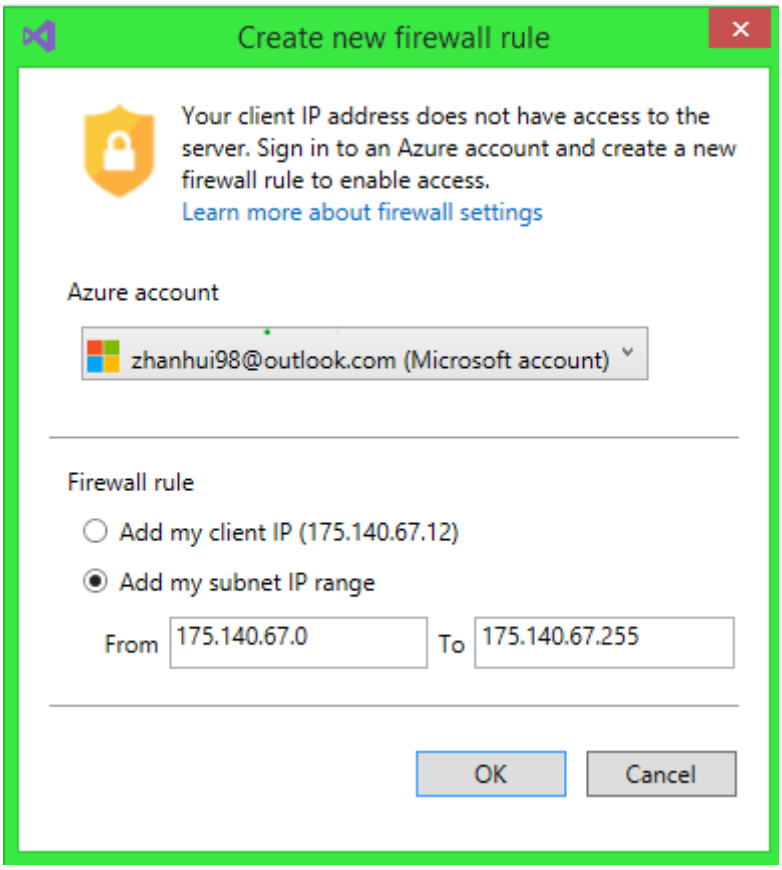


Figure 4.1.4.2 Connect local app to Azure SQL Database

Add connection strings on the App Services

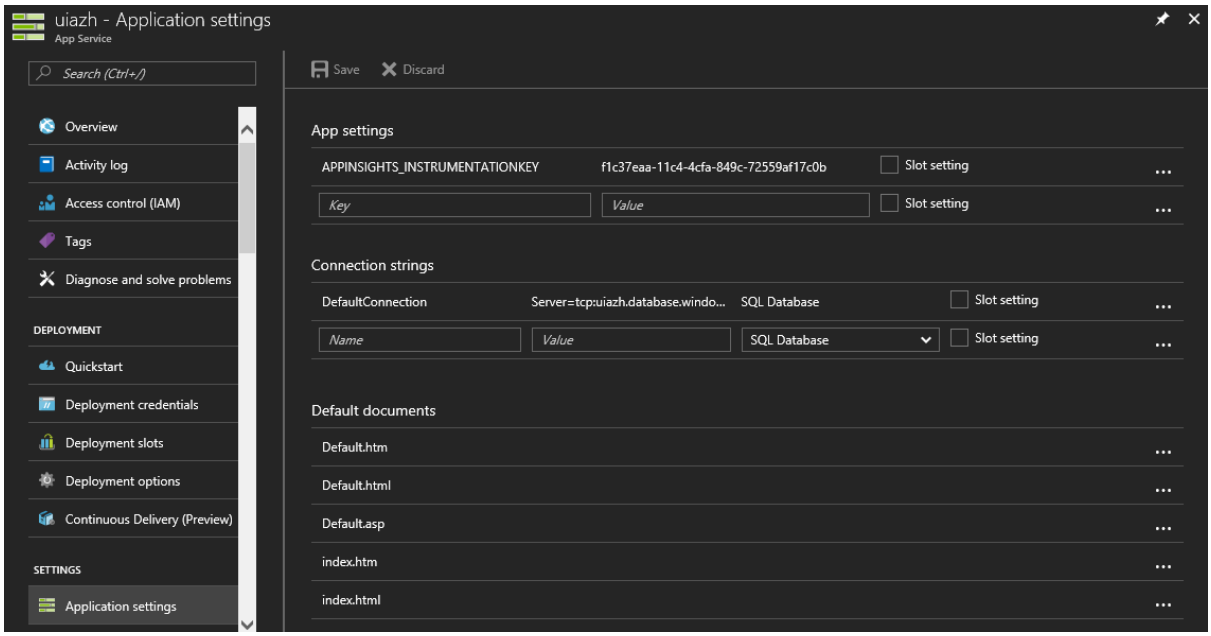


Figure 4.1.4.3 Add the connection strings on the App Services

Deployment of the web app to Azure App Services

As the Web App Service has been created earlier, the which have been covered in section 4.1.2, the application can be published to the Azure. By right clicking the solution on Visual Studio and click on "Publish", select the target which is "Microsoft Azure App Service", the following window will be displayed.

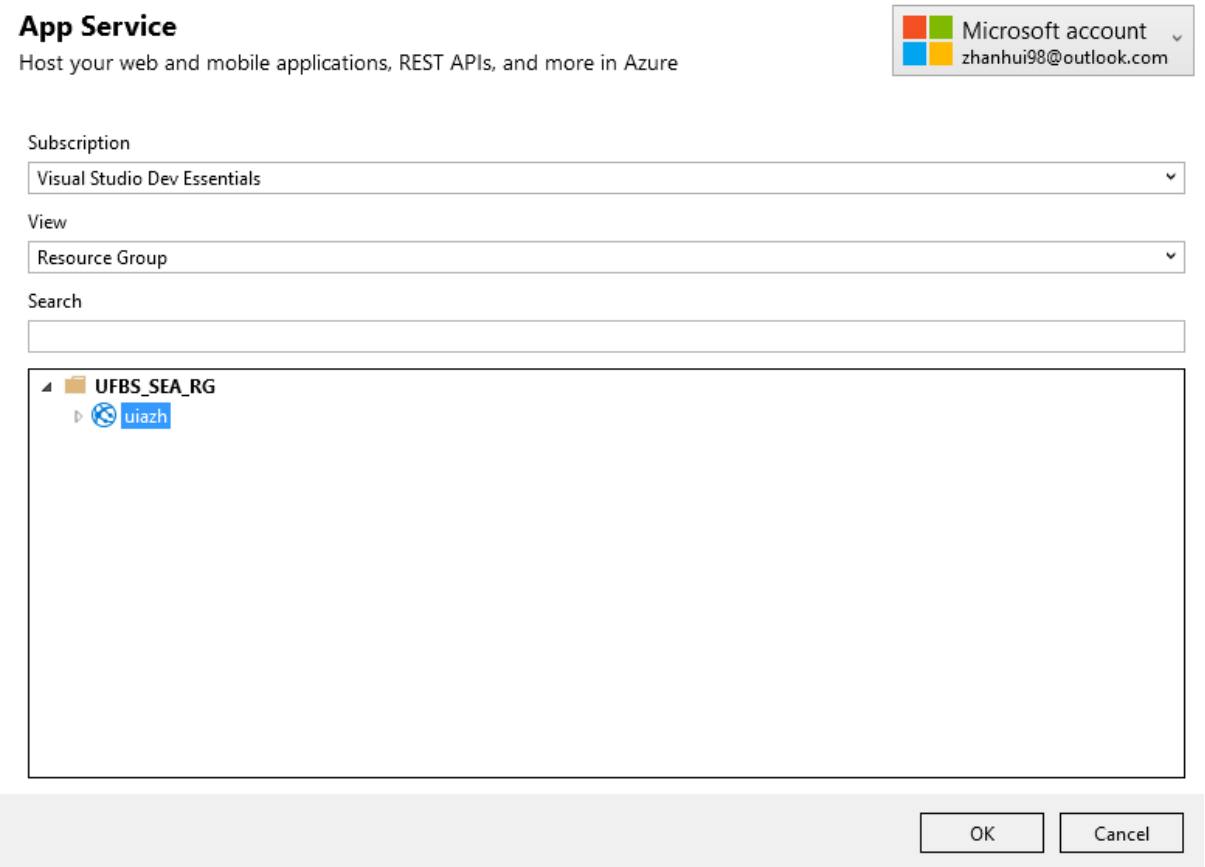


Figure 4.1.4.4 App Service in Southeast Asian Region is selected

Prior to publication, ensure that database connection strings are completed with the login credentials.

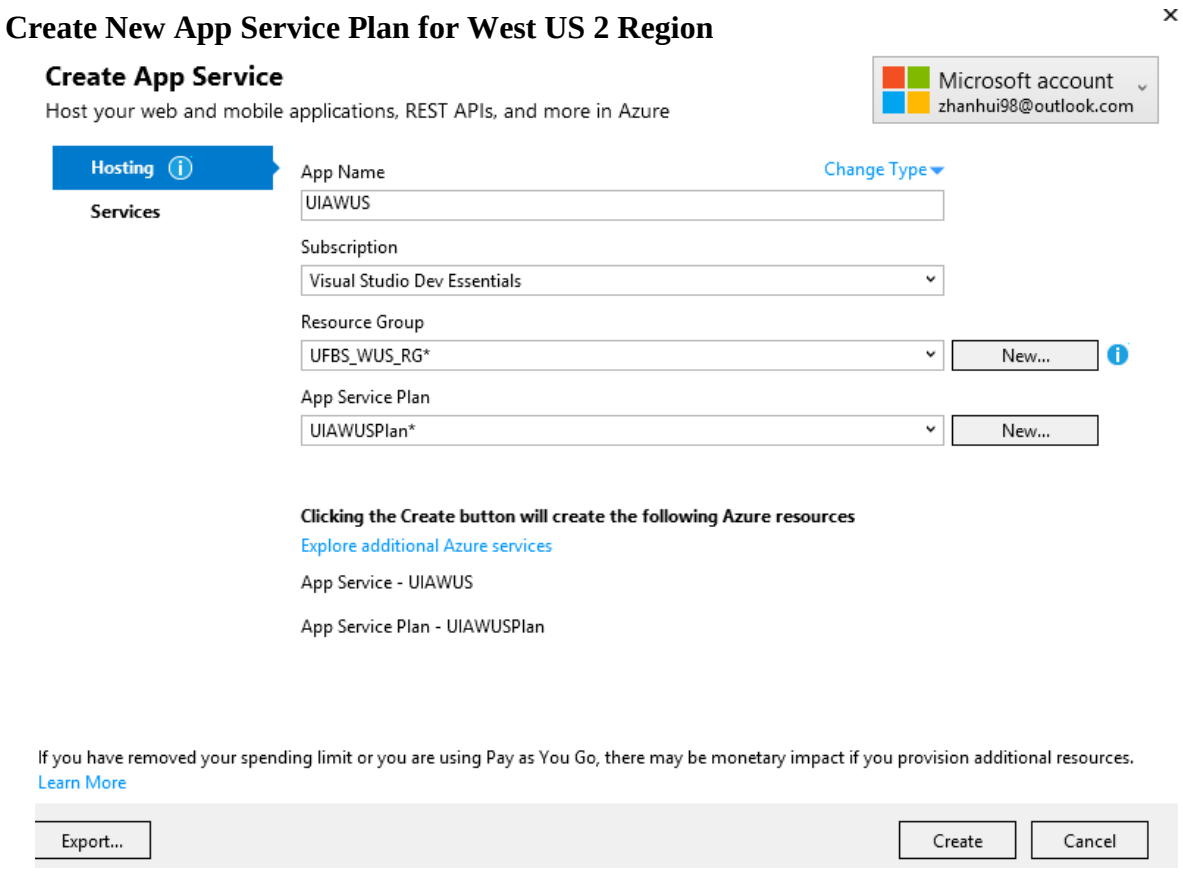


Figure 4.1.4.5 App Service for West US 2 Region

The web application will be published on the second web app which is the hosted on West US 2 region. Both of the web app published to Azure are displayed below.

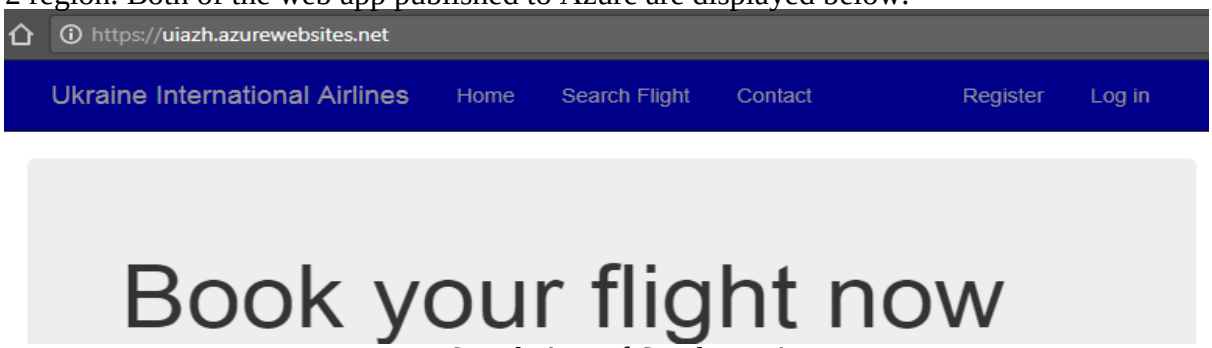


Figure 4.1.4.6 Web App of Southeast Asia Region

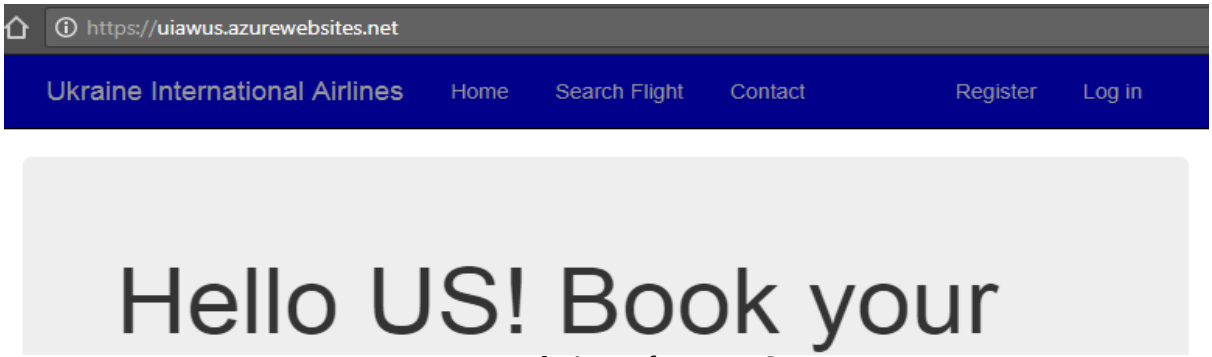


Figure 4.1.4.7 Web App of West US 2 Region

HTTPS only on Azure Web Application

As displayed on the figure 4.1.4.6 and figure 4.1.4.7 the UIA Online Flight Booking websites are redirected on Hyper Text Transfer Protocol Secure (HTTPS) where it provides a more secured connected compared to the conventional HTTP by encrypting the data transmitting on the network. With HTTP the data are transferred in plaintext where it the login credentials can the captured by the attacker easily when there are in the same network. With HTTPS, UIA Online Flight Book System have a better protection on the customers' activities on the website.

The following code is appended on the web.config file to achieve the HTTPS on Azure site.

```
<rewrite>
<rules>
 <rule name="Redirect to https">
  <match url="(.*)"/>
  <conditions>
   <add input="{HTTPS}" pattern="Off"/>
  </conditions>
  <action type="Redirect" url="https://{HTTP_HOST}/{R:1}"/>
 </rule>
</rules>
</rewrite>
```

Figure 4.1.4.8 URL Rewrite Rule

4.1.5 Storage account

In the Azure portal, storage account is created to store the audit logs of the databases.

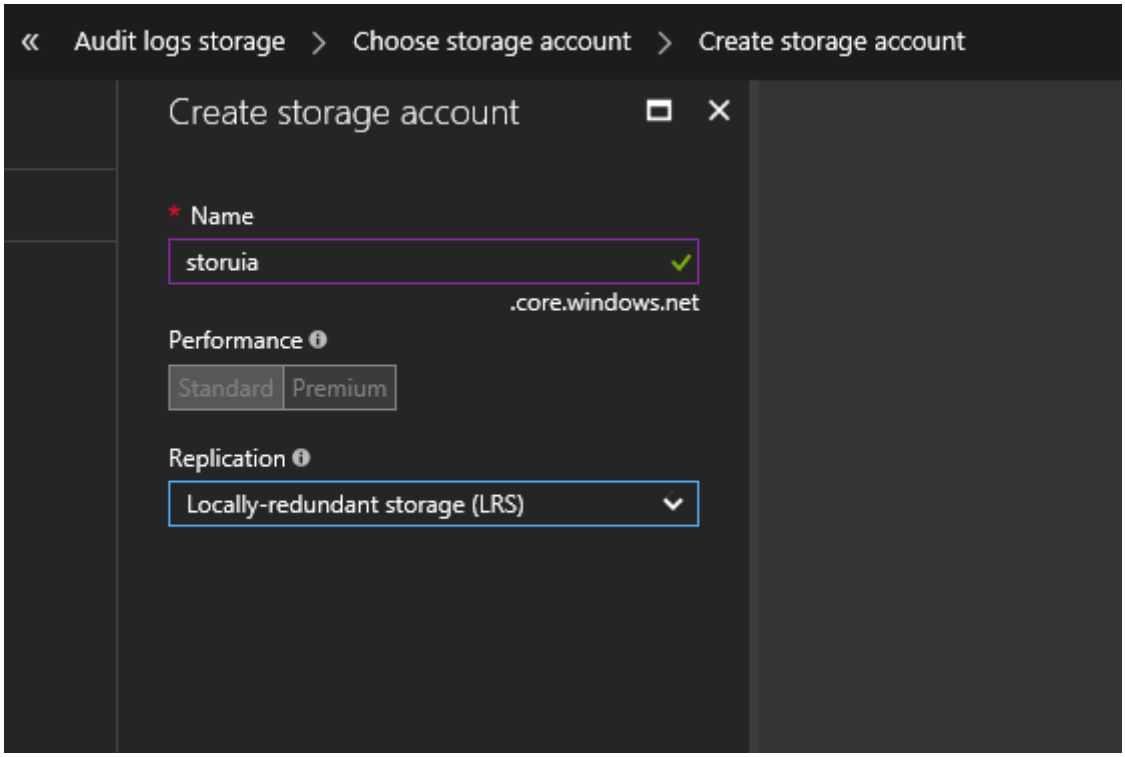


Figure 4.1.5.1 Creation of storage account

Turn on auditing and threat detection

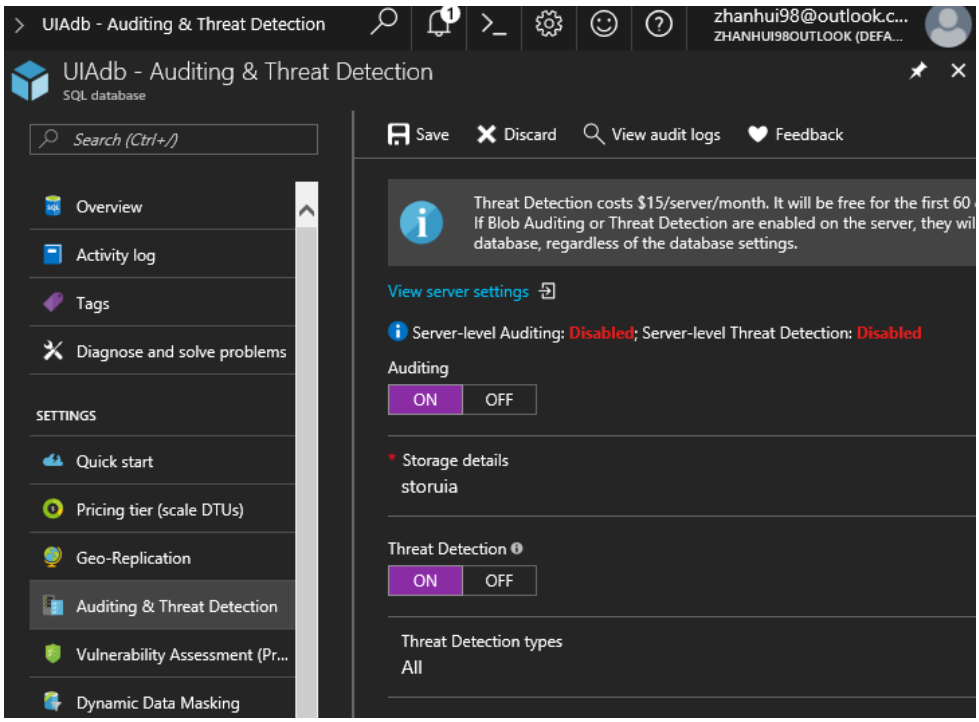


Figure 4.1.5.2 Auditing & Threat Detection

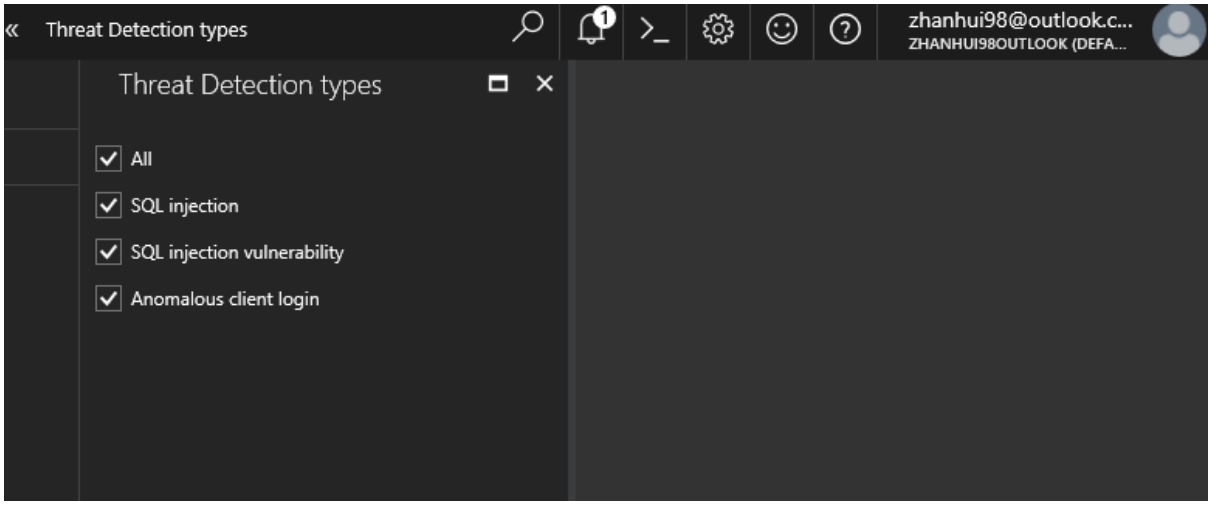


Figure 4.1.5.3 Threat Detection Types

All of the threat detection types includes SQL injection, SQL injection vulnerability and anomalous client login has been selected to detect unusual activities on the database. The logs will be stored in the storage account setup earlier which provides useful information for security analyst of UIA. Glover (2015) states that log management is essential to ensure that there are evidences to trace back in the event of the security breach as well as detected potential attacks.

4.1.6 Traffic Manager

Traffic Manager offered by Microsoft Azure allows the distribution of user traffic in different datacenters to be controlled where there are four traffic routing methods can be selected which includes priority, weighted, performance and geographic (Dwivedi, et al., 2017).

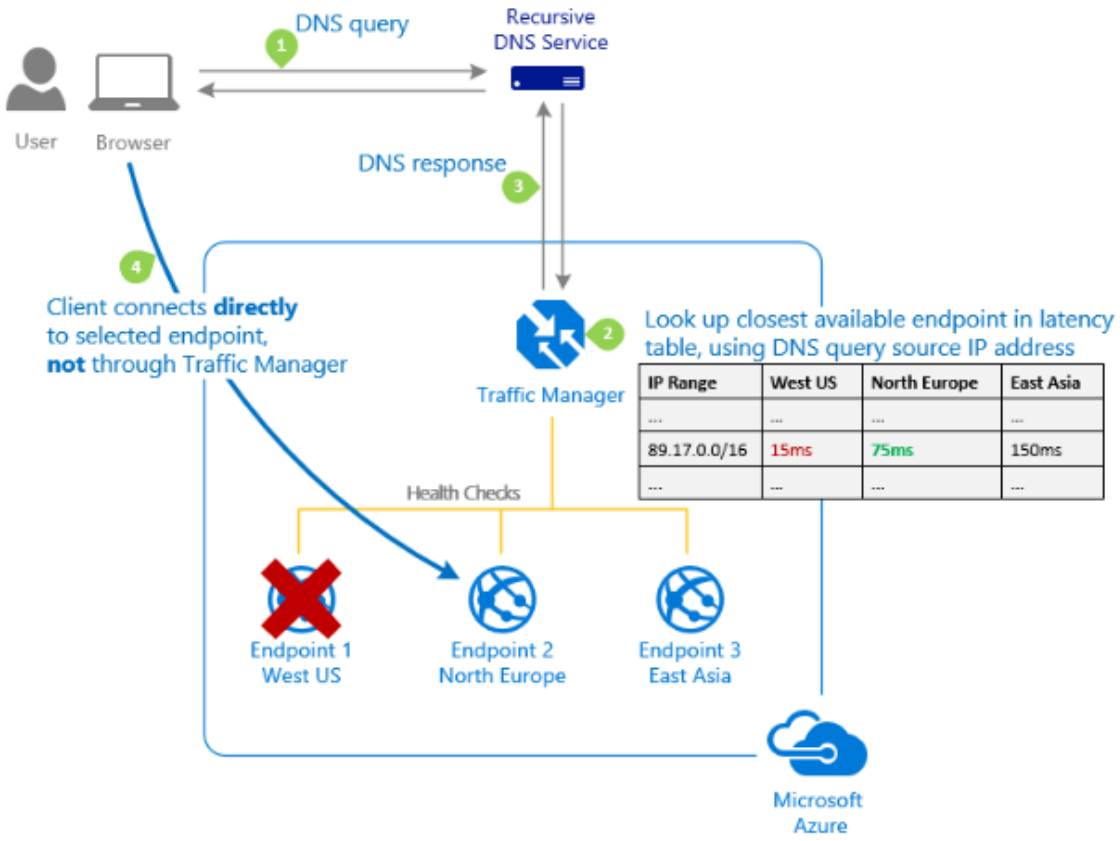


Figure 4.1.6 Performance traffic-routing method (Microsoft, 2017)

In the perspective of the UIA Online Flight Booking System, there are two endpoints in different geographical location which is South East Asia and West US 2. Performance method is selected for the traffic routing to offer high web app performance to the end users which is the customers by routing the users to the endpoint which has the lowest network latency. According to Microsoft (2017), the Performance traffic routing method of the Traffic Manager selects the endpoint in Azure datacenter with the lowest latency on that IP addresses and return the particular endpoint in the DNS response.

Create Traffic Manager Profile

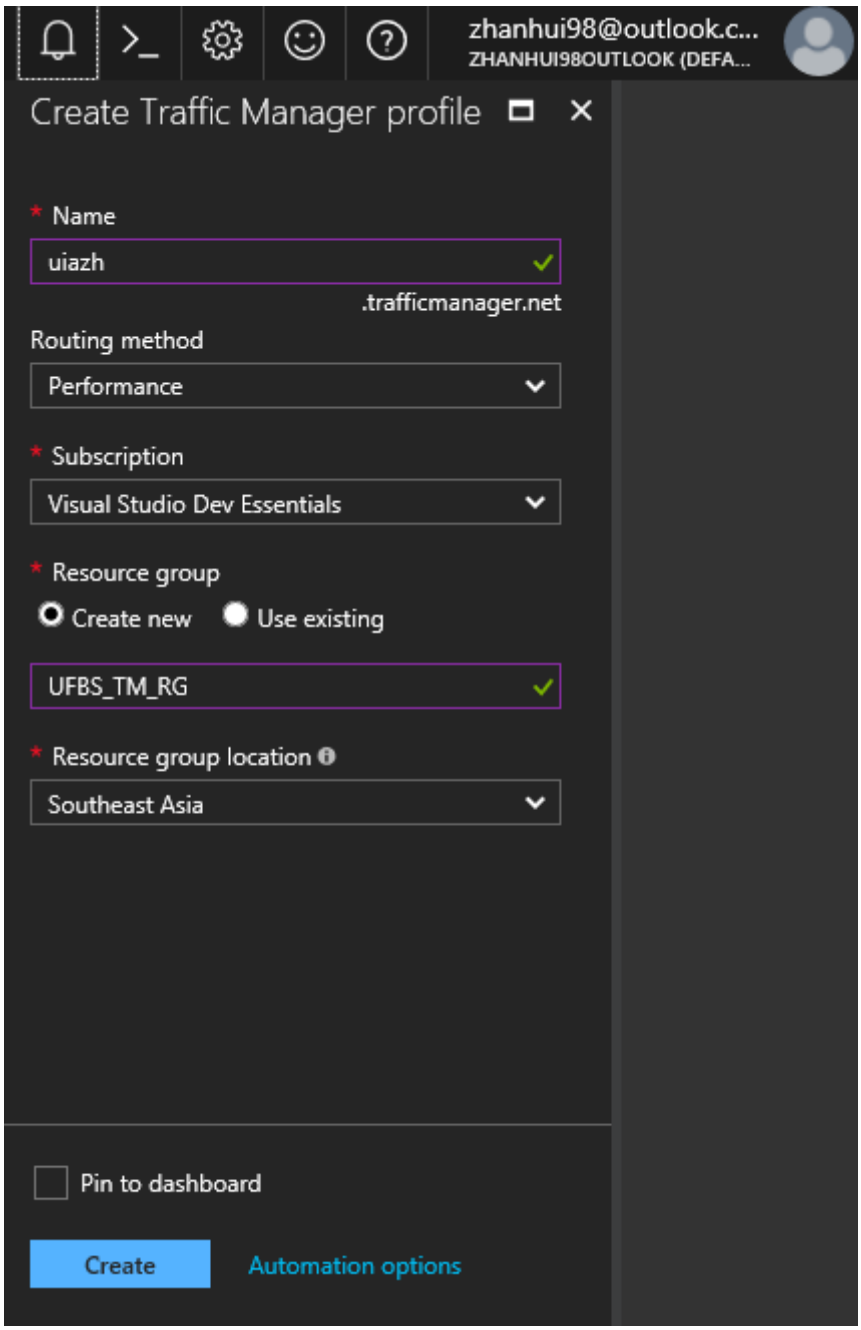


Figure 4.1.6.1 Traffic Manager Profile

The Traffic Manager profile is named “uiaTM”. As the primary focus of UIA is on the performance of the web application, the performance routing method is selected for the UIA Web App. The functionalities of the performance routing method have been described above. The resource group used to hold the traffic manager profile is UFBS_TM_RG.

Configure Endpoints

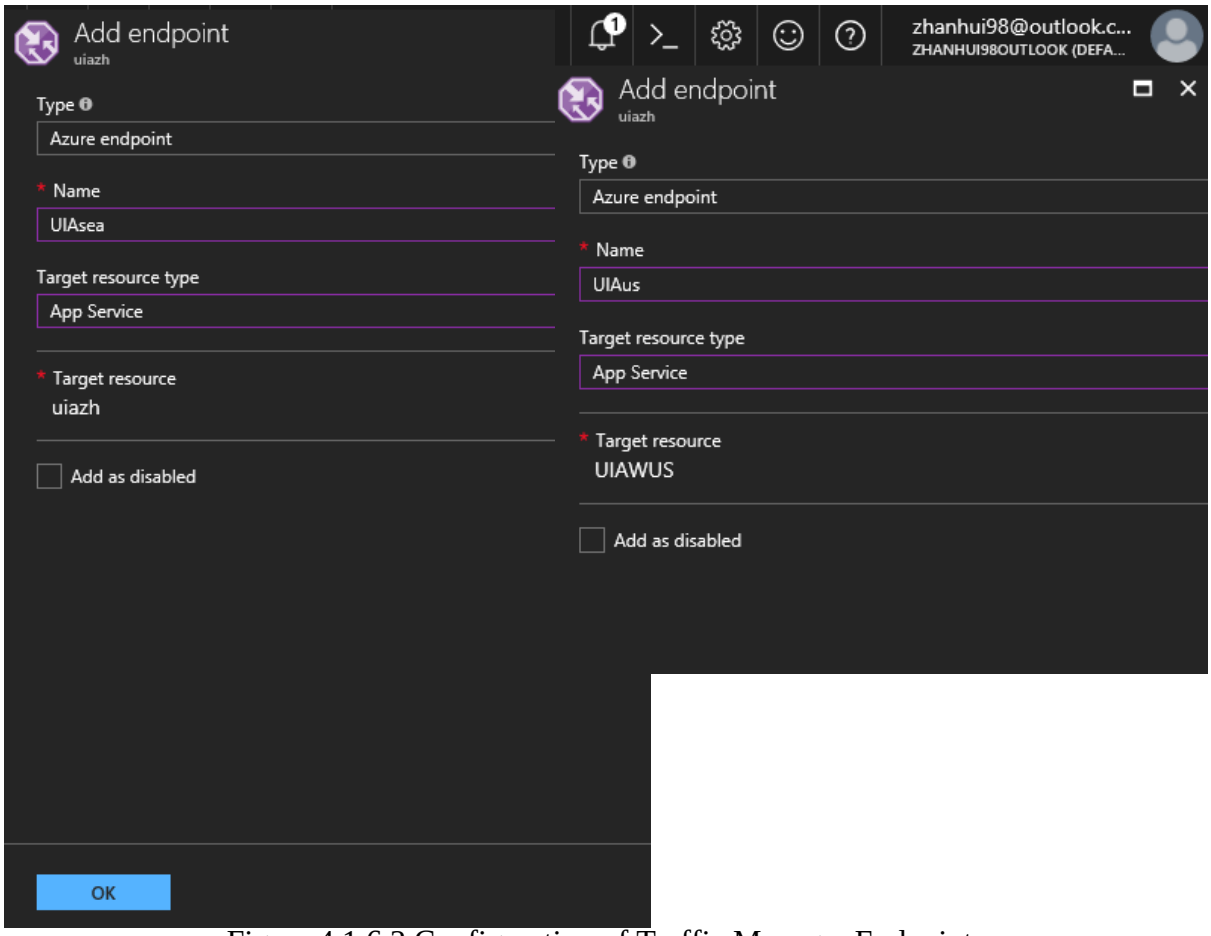


Figure 4.1.6.2 Configuration of Traffic Manager Endpoints

The endpoints are given the name of UIAsea and UIAus respectively where the endpoint type is Azure endpoint and the target resource type is “App Service”. The target source is linked to both App Services in Southeast Asia and West US 2.

The traffic manager endpoints are online after the configurations.

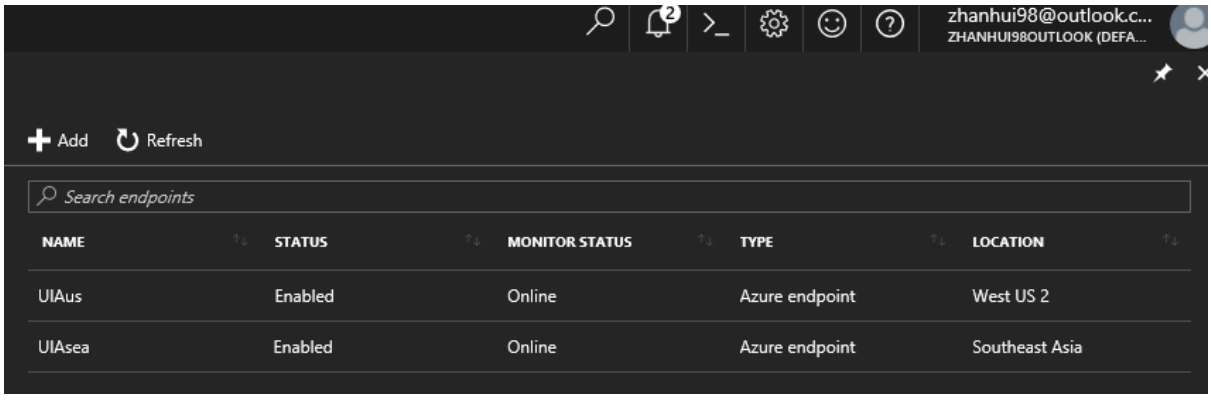


Figure 4.1.6.3 Traffic Manager Endpoints on Southeast Asia and West US 2

Connection Testing from various regions

GeoPeeker is used to check where the App services will be redirected based on the users from various location.

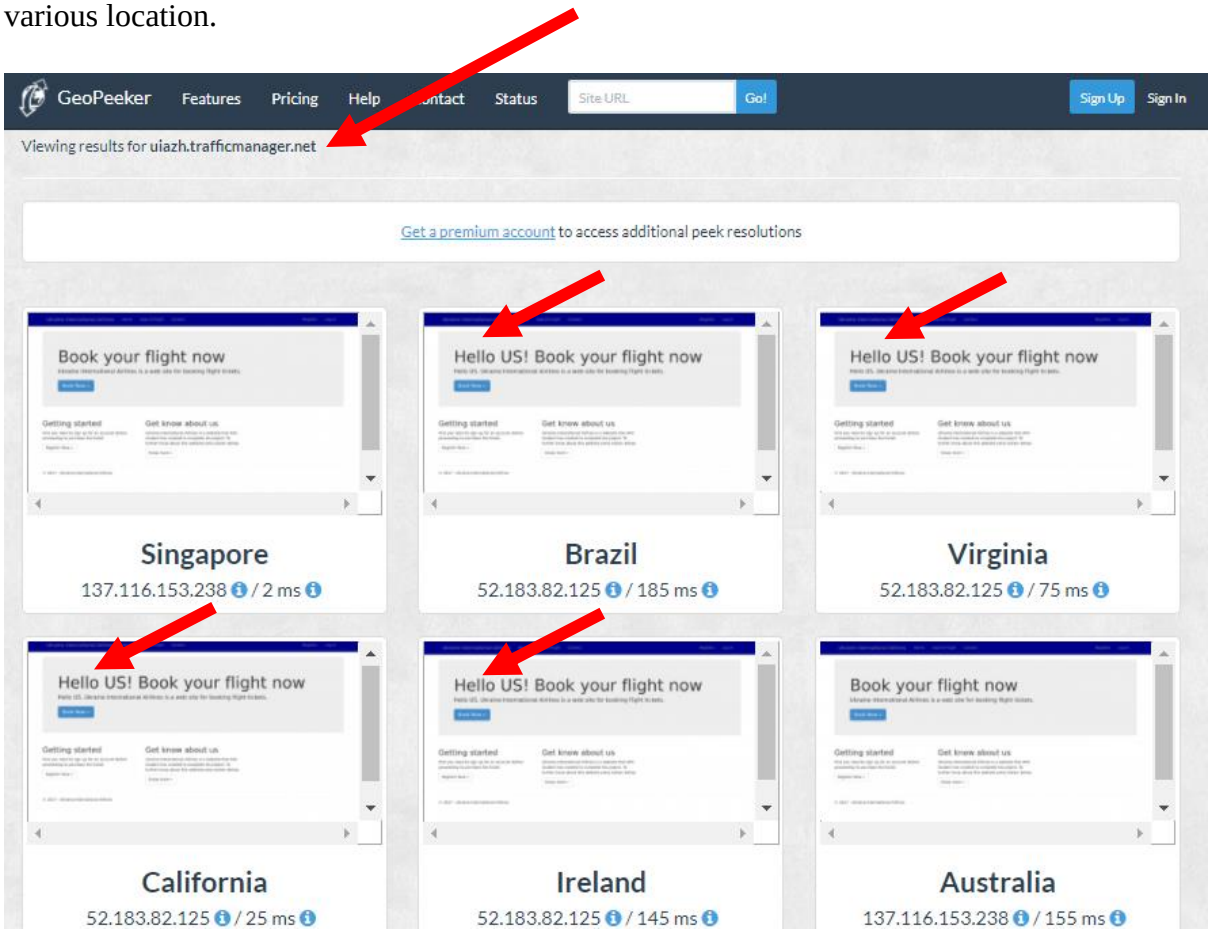


Figure 4.1.6.4 GeoPeeker Results on Traffic Manager Endpoints on Southeast Asia and West US 2

Based on the figure above, the URL of the traffic manager (uiazh.trafficmanager.net) has been tested. The URL visited from Brazil, Virginia, California and Ireland have been redirected to the West US 2 endpoint as the homepage with “Hello US!” is displayed.

GitHub Deployment

Visual Studio Team Services (VSTS), GitHub, Bitbucket are among the platforms available for developers to configure the deployment for the web applications. The figure below displays the Github Authentication and the branches.

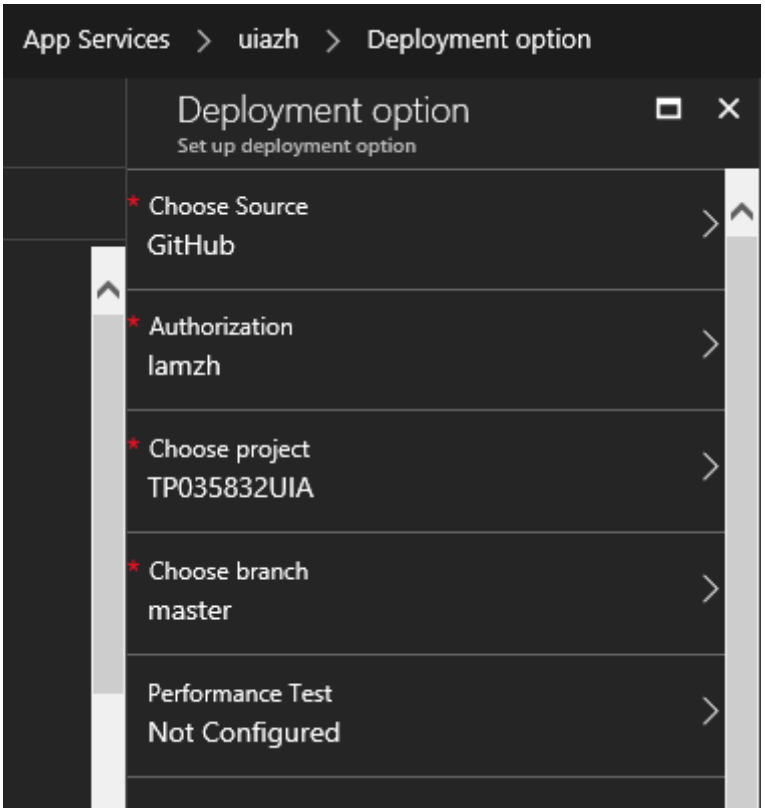
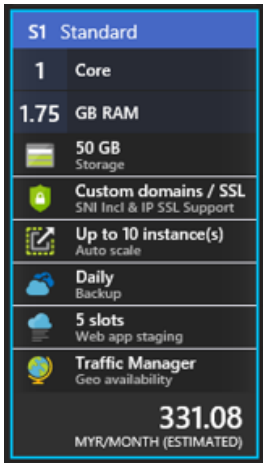


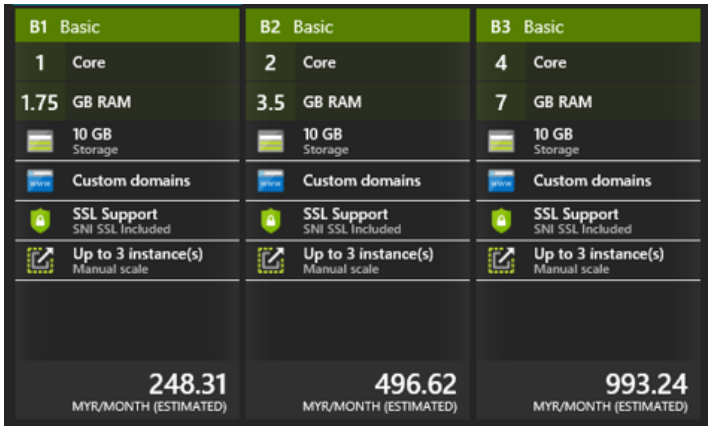
Figure 4.1.6.5 Setting up Deployment Option

4.2 Justification of Selected Plans and Application Scaling

4.2.1 Selected Web App Plan



As mentioned in Section 4.1, S1 Standard is the pricing tier selected for the web application plan. The S1 Standard plan comes with single core, 1.75 GB RAM where it supports custom domain. Apart from that, the S1 Standard package is capable to scale up automatically up to 10 instances which is ideal meet the demand during the peak seasons for UIA. Moreover, traffic manager can be configured on the web application with the selected plan as well.



Basic plan such as B1, B2 and B3 are not considered due to lack of backup features, absence of traffic manager and supports on manual scale up to 3 instance which is significantly inferior to the package offered in Standard pricing tier where the pricing difference is negligible as the estimated monthly difference of B1 Basic and S1 Standard tier is less than RM85.

In the perspective of S2 and S3 Standard pricing tier, the cores and the RAM are the only difference where S2 Standard has dual core for processing along with 3.5 GB RAM whereas, S3 Standard, comes with Quad Core and 7 GB RAM. However, the price of the S2 and S3 Standard plan is roughly twice and four times more expensive than the S1 Standard.

Considering the scale of the UIA, S1 Standard is ideal as it has a greater price-to-performance ratio for the operation of the Online Flight Booking System of UIA considering the extra functions and automation features offered. Therefore, S1 Standard package is selected for the App Service of UIA Online Flight Booking System.

4.2.2 Selected SQL Database Plan

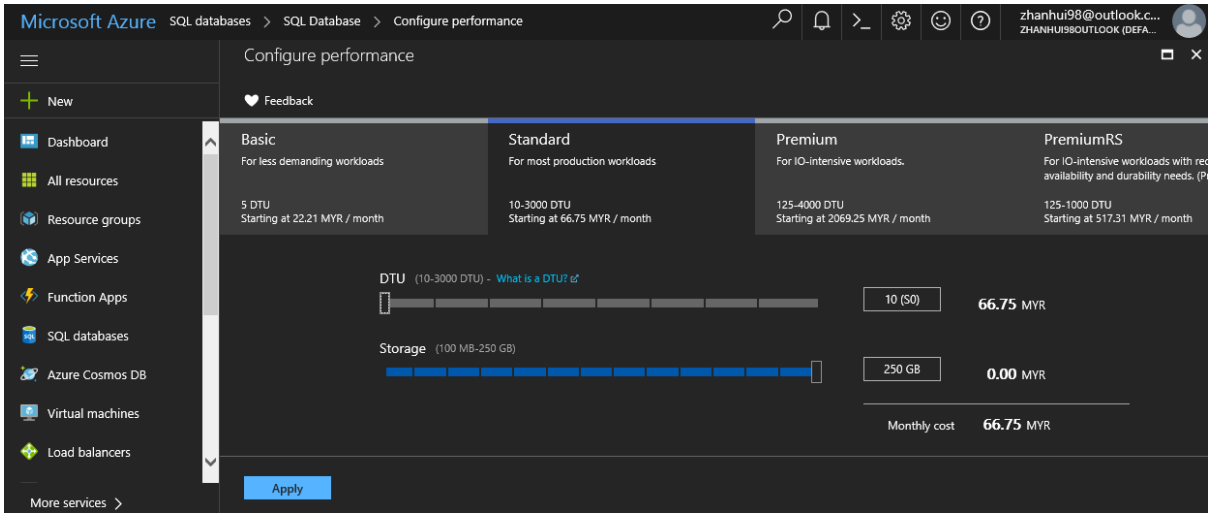


Figure 4.2.2 SQL Database Plan

As mentioned in Section 4.1, S0 Standard is the pricing tier selected for the database. The database size and the number of data throughput unit (DTU) and the database size are the primary considerations while selecting the pricing plan for SQL database, rather than the CPU power. In terms of the basic plan, it offers 5 DTUs and 2GB of database. Allen (2017) suggests that basic service plan is ideal testing purpose only. Considering UIA is expanding of its flight operation across the world, S0 pricing tier is selected with 250GB of database size for the scalability of database.

4.2.3 Application Scaling

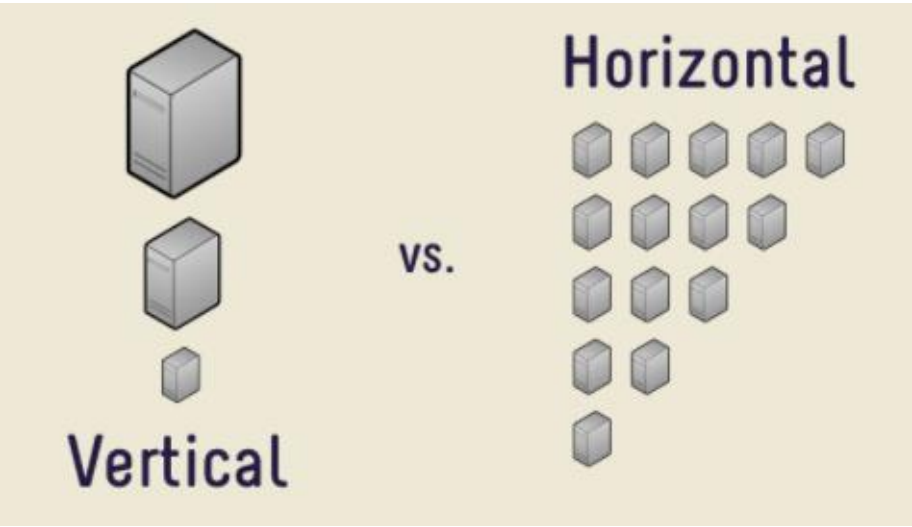


Figure 4.2.3.1 Vertical Scaling (Scale-Up) and Horizontal Scaling (Scale-Out) (Georgiev, 2014)

App Services in Azure can be scaled in two ways, which includes scale-up and scale-out. Sreeram (2016) states that scale-up, also known as vertical scaling, is to upgrade the capacity of the host by expanding the cores amount and memory whereas scale-out, horizontal scaling, is to increase the number of instances where multiple identical copies of the website are distributed with the load balancer.

In the perspective of the UIA Online Flight Booking App Service, it is configured for scaling out. Microsoft Azure supports automatic horizontal scaling. The image below displays the scale out configuration.

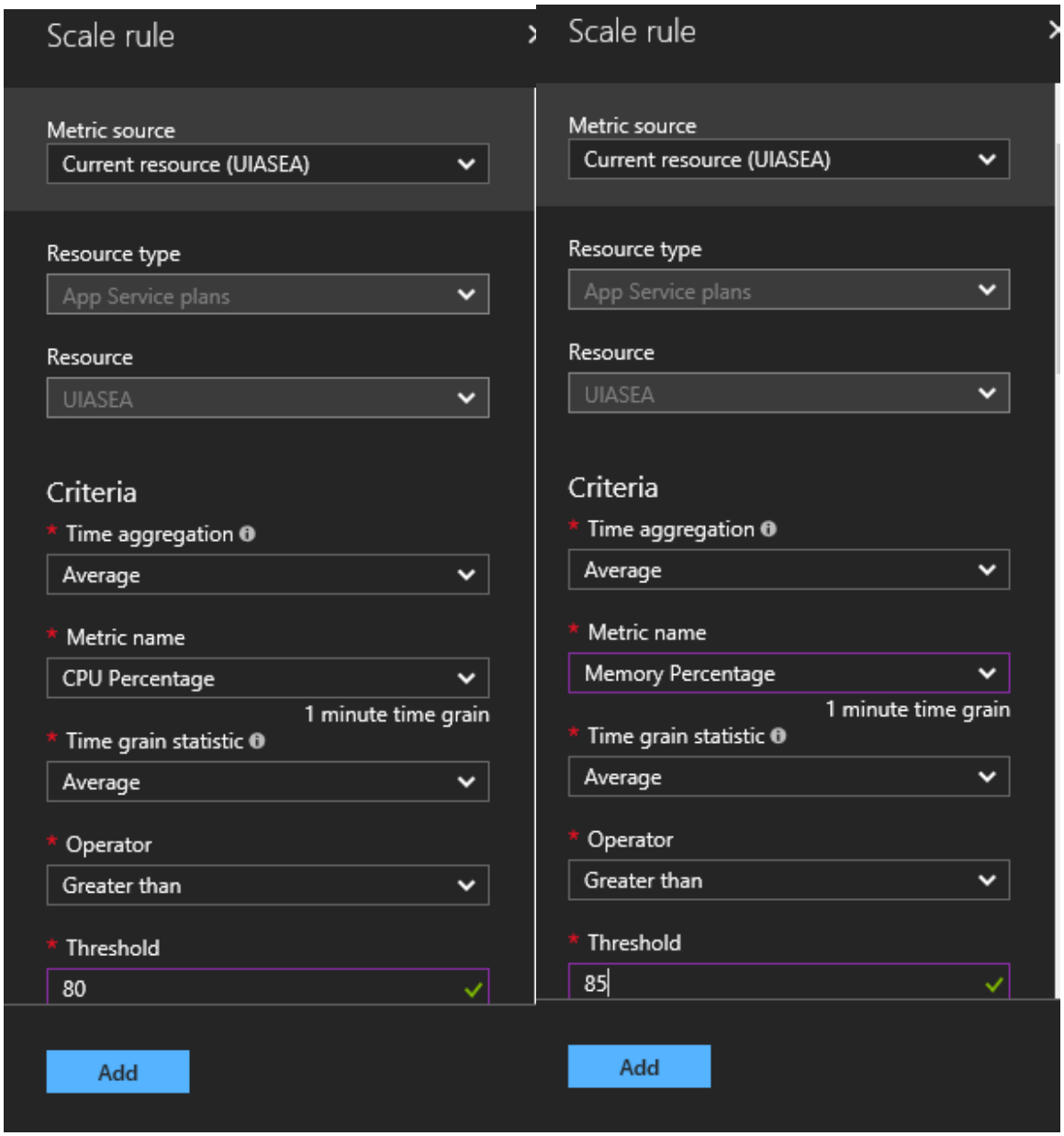


Figure 4.2.3.2 Scale Rules of the App Service in Southeast Asia Region

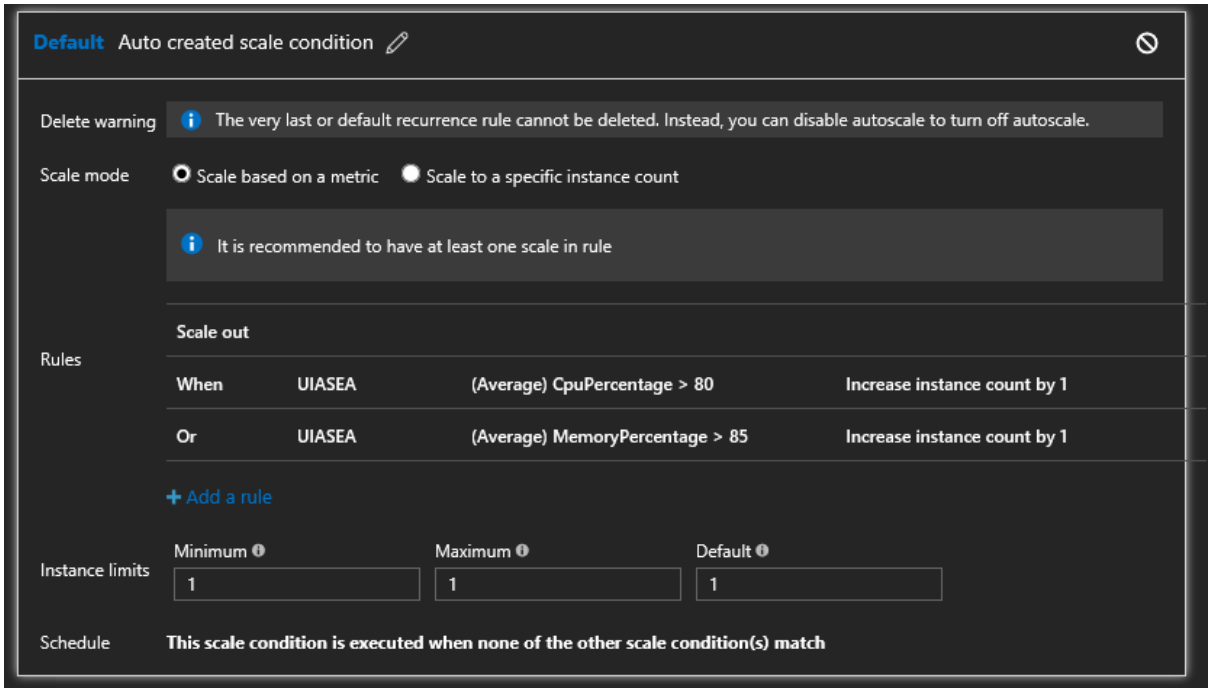


Figure 4.2.3.3 Configured Auto Scaling condition

The scale rule for the UIA Online Flight Booking Web App is based on metric source which is the app service plan. CPU percentage, disk queue and memory percentage are among the available metric options. The scale rule for the web app includes the instance of the web application will expand by 1 when the CPU load is above 85% for more than 10 minutes. Based on the selected pricing tier, S1 standard, the Web App is capable to scale out up to 10 instances.

5.0 Test Plan and Testing Discussion

Software testing is an essential part of the system development to ensure that the bugs are identified prior release as a complete version to end users (Saini & Rai, 2013). A well-organized test plan can boost the success rate of a project. Out of many types of testing, unit testing tests the individual units of the system, such as a system's function, which is the lowest level of software testing while performance testing tests the stability, responsiveness and the speed of the system (Lepistö, 2012).

As the testing requires Visual Studio Team services to manage and collaborate online. In the circumstance of coding in industrial environment, there will be plenty of peoples in the team to work on the project therefore it is not ideal to run from the local machine.

To perform the testing, Visual Studio Team Services (VSTS) is required for managing and collaborating on the project online. In the perspective of the coding project in industry, there will be teams of developer working on the same project. Hence, VSTS is ideal for collaboration rather than running from local machine.

The following pages are the test plan for the unit testing as well as the testing discussions of the performance test on the UIA Online Flight Booking System which is performed on Microsoft Azure.

5.1 Unit Testing

Test Case ID	OFBS01				
Module Name: Customer Registration					
Pre-conditions: Customer does not has an account UIA Online Flight Booking System					
Case No.	**Test Scenario**	**Expected Output**	**Actual Output**	**Remarks**	**Comment**
1.	Incomplete customer information with invalid email	Error Message will be displayed	Error Message "Invalid details" displayed	Pass	-
2.	Complete customer information with invalid email	Error Message will be displayed	Error Message "Invalid Email" displayed	Pass	-
3.	Complete customer information with valid email and mismatched password **Email**: someone@hotmail.com **Password**: P@ssw0rd **Renter Password**: password	Error Message will be displayed	Error Message "Password Matched" displayed	Pass	Good password complexity
4.	Complete customer information with valid email and matched password **Email**: someone@hotmail.com **Password**: P@ssw0rd **Renter Password**: P@ssw0rd	Successful Registration	Message "Successful Registration" displayed and user will be redirected to Login Page	Pass	-

Test Case ID	OFBS02					
Module Name: Customer Login						
Pre-conditions: Customer are not logged in to the UIA Online Flight Booking System						
Test No.	**Username**	**Password**	**Expected Output**	**Actual Output**	**Remarks**	**Comment**
1.	Empty	Empty	Error Message will be displayed	Error Message "Fields should not be empty" displayed	Pass	-
2.	Invalid	Invalid	Error Message will be displayed	Error Message "Invalid Login Credentials" displayed	Pass	-
3.	Invalid	Valid	Error Message will be displayed	Error Message "Invalid Login Credentials" displayed	Pass	-
4.	Valid	Invalid	Error Message will be displayed	Error Message "Invalid Login Credentials" displayed	Pass	-
5.	Valid	Valid	Login successful	Page with the user account is loaded	Pass	-

Table 5.1.2 Test Case For Customer Login

Test Case ID	OFBS03				
Module Name: Flight Searching					
Pre-conditions: Visitors and customers should have access to UIA Online Flight Booking Website					
Case No.	**Test Scenario**	**Expected Output**	**Actual Output**	**Remarks**	**Comment**
1.	Visitors and customers search the trip which is offered by UIA	Related trip will be displayed	Related trip displayed according to the search criteria	Pass	-
2.	Visitors and customers search the trip which is not offered by UIA	Message will be prompted to inform users on the availability of the trip	Message "Trips not available" prompted	Pass	-

Table 5.1.3 Test Case for Flight Searching

Test Case ID	OFBS04				
Module Name: Flight Booking					
Pre-conditions: Customers have registered account on UIA Online Flight Booking System					
Case No.	**Test Scenario**	**Expected Output**	**Actual Output**	**Remarks**	**Comment**
1.	Customers perform flight booking without login	Login Page prompted	Login Page prompted	Pass	-
2.	Customers perform flight booking when logged in	Flight booking performed successfully	Flight booking performed successfully	Pass	-

Table 5.1.4 Test Case for Flight Booking

5.2 Performance Testing

The performance test conducted on Azure Cloud Services only involves the scope of User Tier and Presentation Tier. There are 5 iterations to be used for to perform the test on the web application of the UIA Flight Booking System. 250, 350, 450, 500 and 600.

The performance testing emphasizes on the user and presentation tier where the landing page of the application is tested which is the home page. 5 iterations have been tested where the user load are 250, 350, 450, 500 and 600 respectively with the duration of 5 minutes.

The figure below shows the configuration on one of the performance testing.

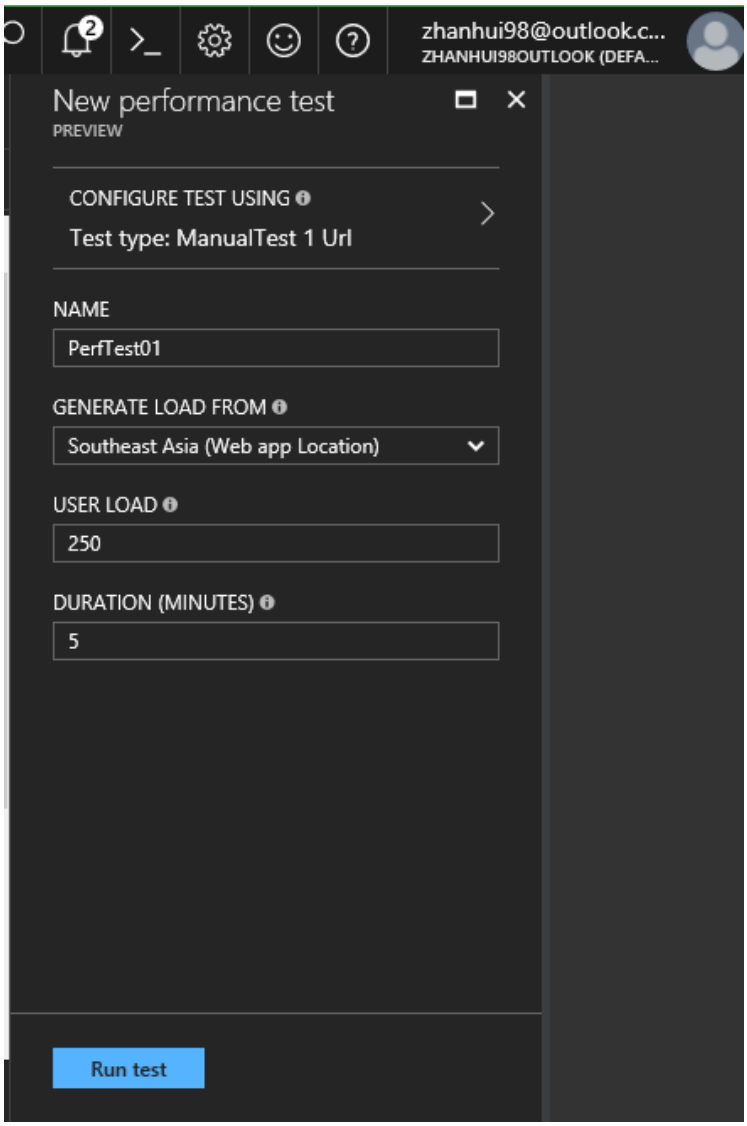


Figure 5.2.1 Setup for performance testing

The test type is manual where the landing page of the website is being tested. “PerfTest01” is the name for the performance test and the 250 user loads are from Southeast Asia which lasted for 5 minutes.

The selection of the testing ranging from the user load of 250 to 600 with a duration of 5 minutes is under the consideration of the available virtual user minutes available every month which is 20000 virtual minutes. The table below displays the calculation on the virtual user minutes to accomplish the performance testing.

User Load	Minutes	Total
250	5	1250
350	5	1750
450	5	2250
500	5	2500
600	5	3000
	Total Virtual Minutes	10750

Table 5.2.1 Virtual Minutes used for the Performance Testing

The figure below displays the results of the performance test.

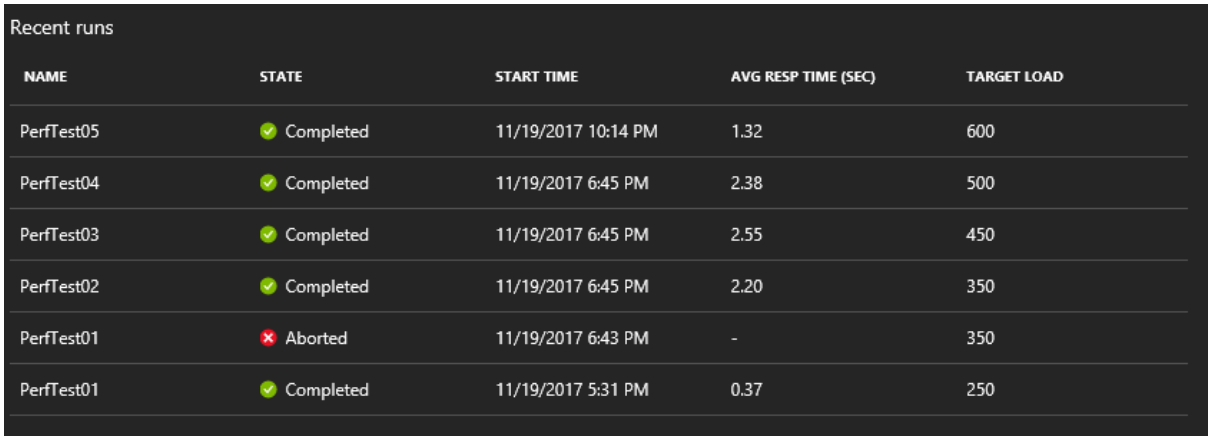


Figure 5.2.2 Performance Testing on the landing page of the UIA Online Flight Booking System

The test results will be displayed in a pie chart where the successful and failed requests are displayed on it.

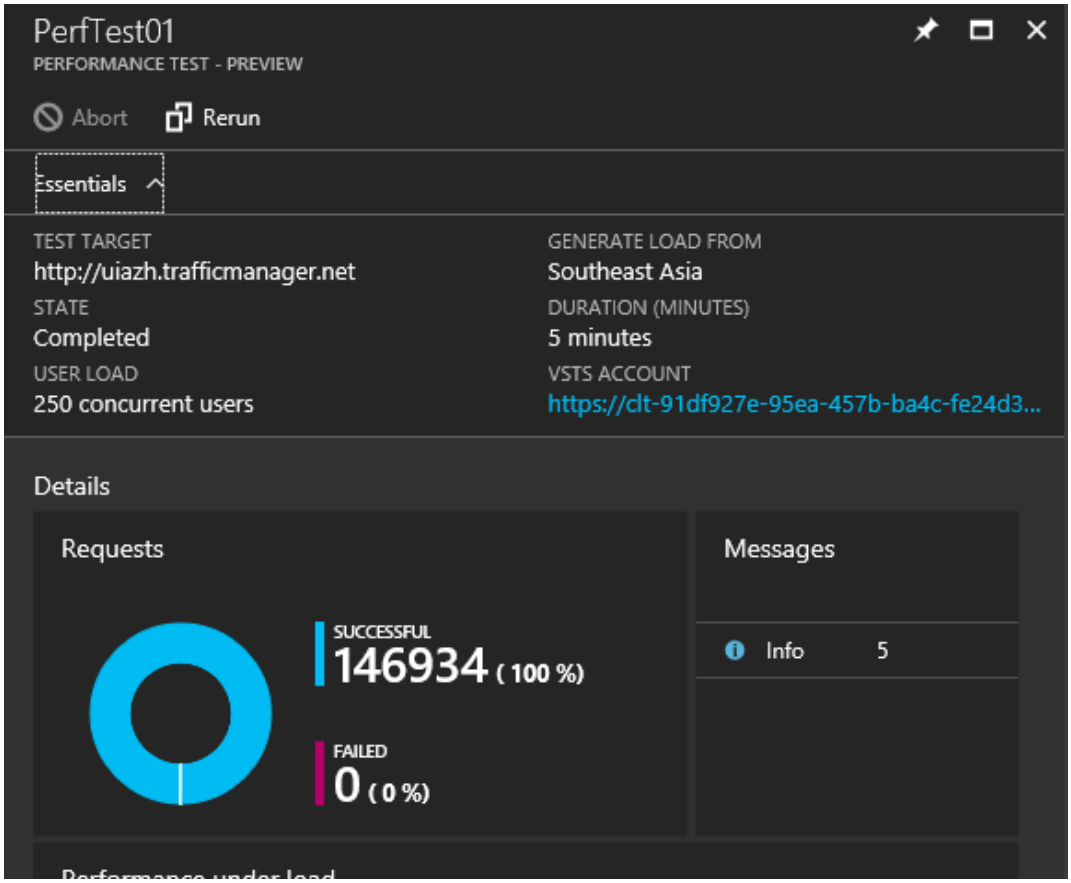


Figure 5.2.3 Performance Test Preview (User Load 250)

Based on the pie chart above, in the test with 250 user loads, 146934 requests were sent and 100% of the requests were successful.

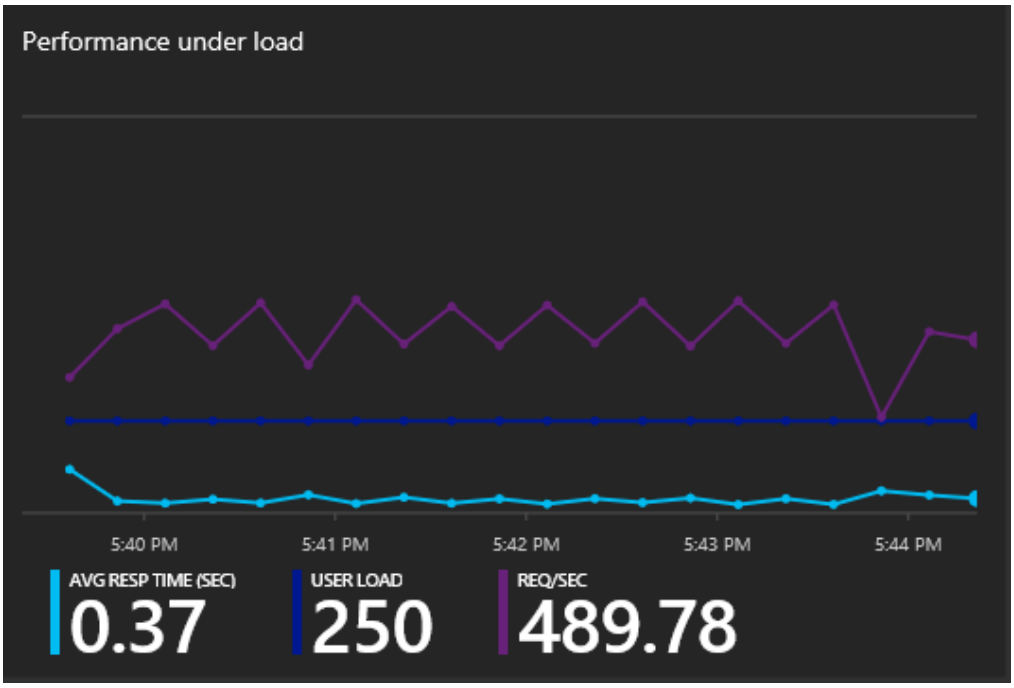


Figure 5.2.4 Performance under user load of 250

The average response time is 0.37 seconds where the web application serves 489.78 requests per second in the period of 5 minutes. The response time is acceptable when the user load is at 250.

The table below shows the results of the performance testing of user load around 350 and 500.

User Load	Average Response Time (sec)	Request Per Second	Successful Request
250	0.37	489.78	146934
350	2.2	159.74	47921
450	2.55	180.31	54092
500	2.38	217.08	65124
600	1.32	476.11	142834

Table 5.2.2 Virtual Minutes used for the Performance Testing

Based on the information of performance testing gathered on table 5.2.2, the average response time of the landing page is around 0.37 seconds when the user load is at 250 with 146934 successful request is acceptable. The data collected with the user load of 350, 450 and 500 have a longer response time which hovers around 2.5 seconds as the tests have been conducted concurrently. Nevertheless, even at a user load of 600, the average response time is less than 1 second longer than the average response time when the user load is at 250. Based on the testing conducted, the UIA Online Flight Booking System is proven to be sufficient to on the operation scale of Ukraine International Airlines.

More in depth performance testing will be conducted if there are more virtual minutes provided by Azure to allow the developer perform a longer duration of performance testing to discover the limit of the web application.

Web App Usage

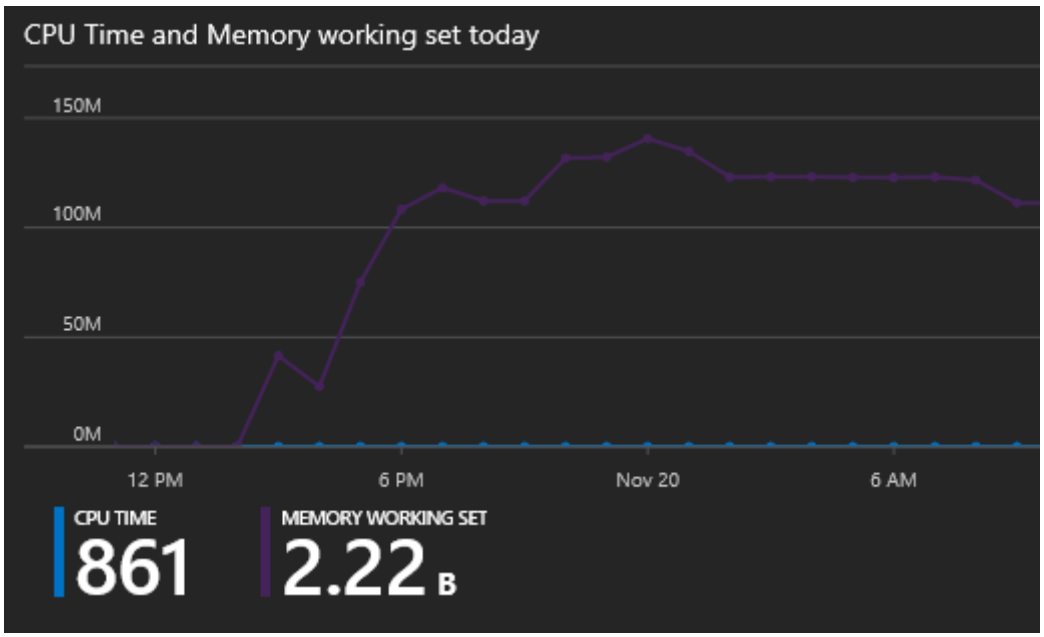


Figure 5.2.5 Live CPU Time and Memory Working set

The figure above displays the usage statistics of the web application for the day. It is a customizable chart that allows other metrics to be included

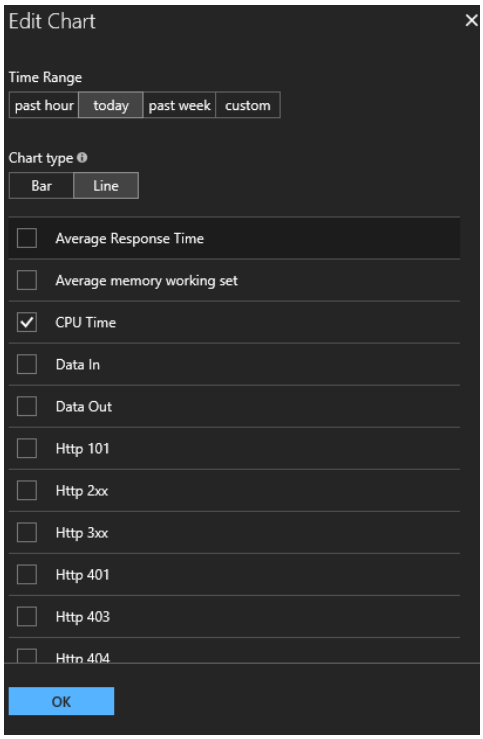


Figure 5.2.6 Customizable metrics on the live chart

This is the chart after adding in the metrics of average response time and HTTP server errors.

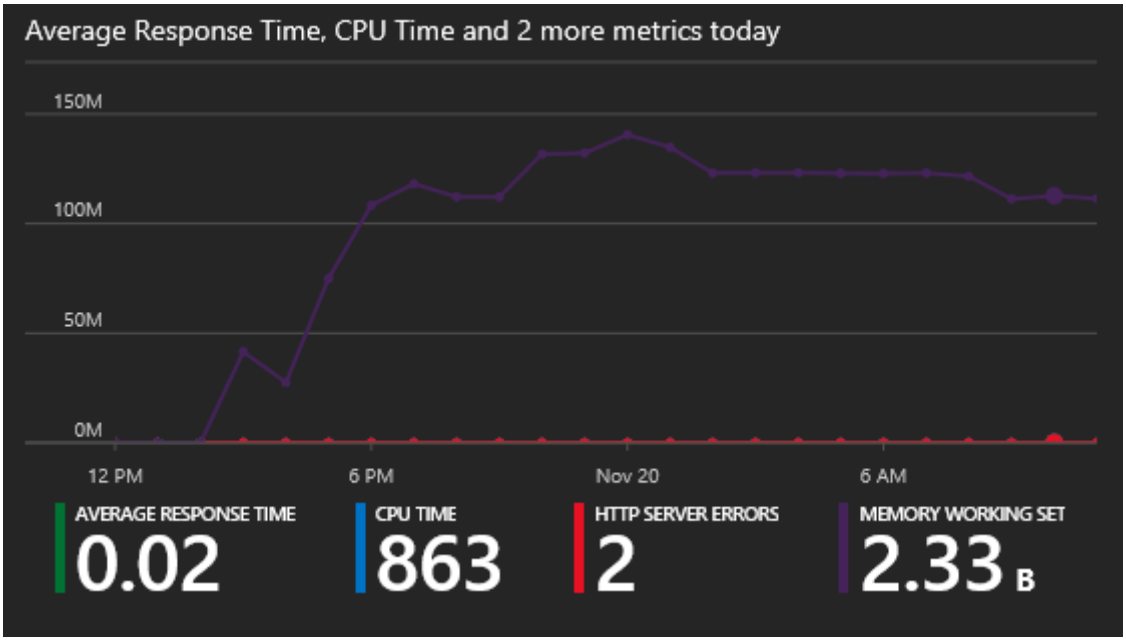


Figure 5.2.7 Customized chart of Web App Usage

On the other hand, the alert rules can be created based on the monitoring metrics and configured the trigger of the alert to notify the authorities with email or Webhook.

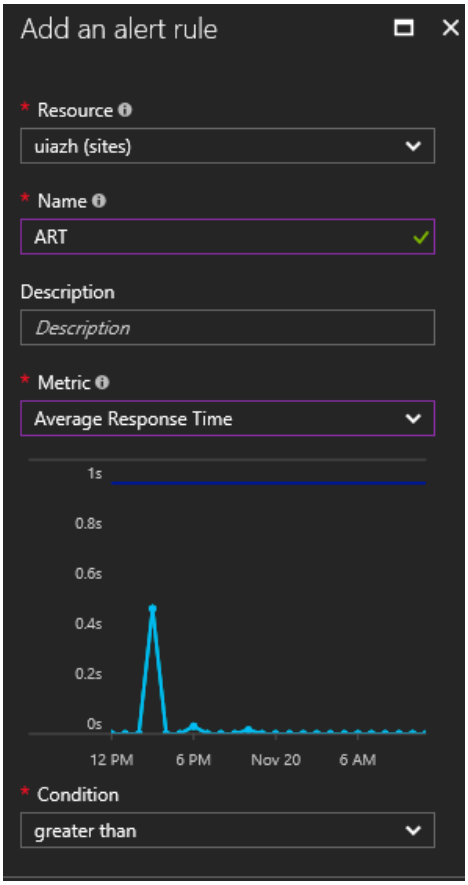


Figure 5.2.8 Alert Rule on Average Response Time of the Web Application

6.0 Discussion of Managed Databases (PaaS)

According to Microsoft (2017), Platform as a Service (PaaS) is a comprehensive development and deployment environment in the cloud that enables a developer to deliver cloud-based application by purchasing resources on a pay-as-use-use basis. Azure SQL Database is a database native to cloud which is also known as Platform as a Service (PaaS) or Database as a service (Rabeler, et al., 2017).

By implementing PaaS on UIA Online Flight Booking System, the hardware costs and administrative costs can be reduced where the SQL Database is built on standardized hardware and software owned by Microsoft (Rabeler, et al., 2017). There are scaling options available such as scale up and scale out without interruptions. Hence, the efficiency of the operation is enhanced and the paying the required cost with the scaling options provided by Azure.

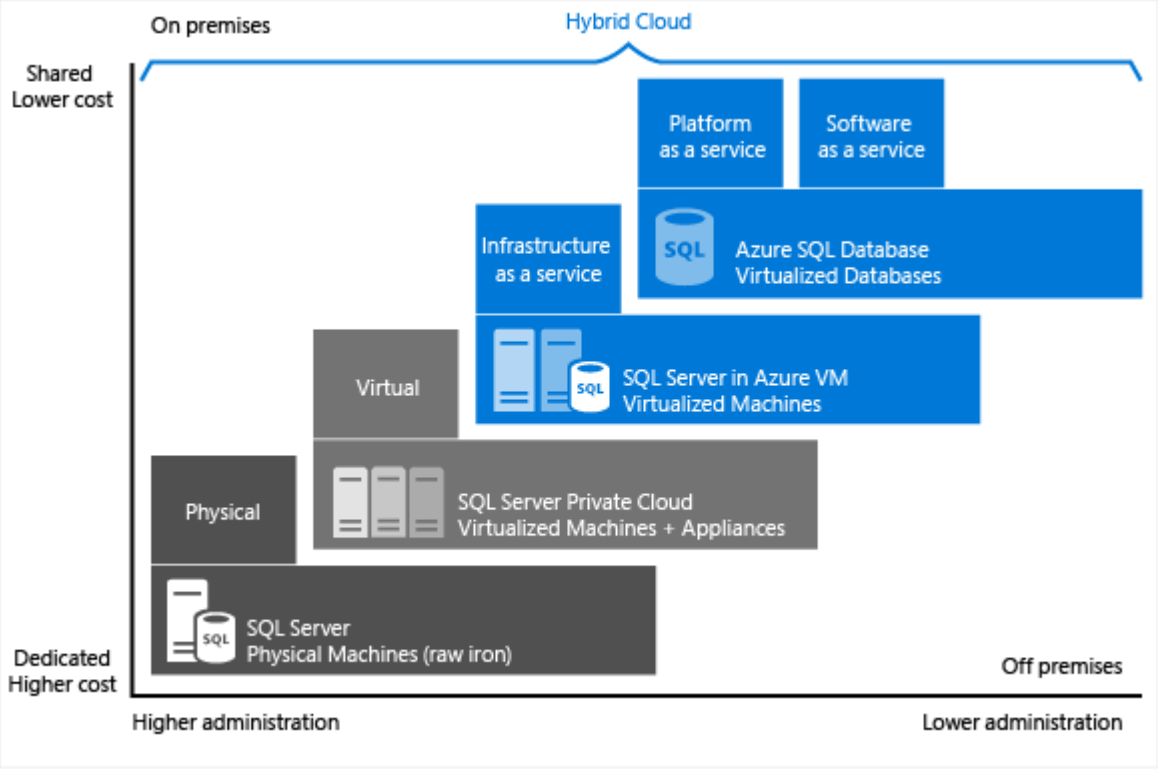


Figure 6.0 Microsoft's Data Platform (Rabeler, et al., 2017)

The advantages of implementing PaaS Azure SQL database over the local server includes cost reduction of provisioning and managing multiple databases. As the administrative cost can be diminished by eliminating the needs of managing operating systems and databases software which includes the activity of upgrading and backup of the databases. With the Azure SQL

Database, the configuration, patch management and upgrading activities are handled by the provider, which is Microsoft. Moreover, the built-in backup capabilities can assist an organization to achieve momentous cost savings especially dealing with a large amount of databases (Rabeler, et al., 2017). The total cost of the application includes the software development cost, SQL Database service costs and a minimal amount of administration cost.

In brief, PaaS database possesses the strengths in terms of the process of application development, deployment and maintenance compared to traditional database. As using the cloud service can assist a cloud-based application to have shorter initial-time-to-market and achieving high Service Level Agreement (SLA) for with the common management operations handled by Microsoft.

7.0 Conclusion

The objectives of the project have been achieved where the Online Flight Booking System of UIA has been designed and developed and published on Microsoft Azure as an App Service. Apart from that, the web application is capable of cater the high demand during the peak seasons by scaling out automatically which is a feature offered in Azure App Service of the selected plan. On the other hand, the web application has a high availability and have a layer of protection on the disasters that might occurred by having the replicated database on the different region to ensure the business continuity is maintained in the event of disasters.

Furthermore, the Application Insights is configured to monitor the health of the application. Also, the project is developed along with standard practices by enhance the maintainability of the project with continuous deployment workflow on GitHub.

Apart from that, security elements have been included in the UIA Online Flight Booking Web App which includes the Auditing & Threat Detection as well as HTTPS URL on the Web App to protect customers' activities on the website.

The UIA Online Flight Booking cloud-based web application on Azure should be able to accommodate the UIA's business needs of expanding into new markets while reducing the operating costs yet effectively managing the flight operations. By adhering to the cloud design patterns, the web application for the UIA can be utilized for a significant period without the need of performing major transitions on the web application.

In brief, the developer has benefitted in the perspective of the cloud computing throughout the development of the project and believe that cloud platform is the upcoming trends for the web applications. The knowledge attained from this module will allow the developer to be better equip to made the contributions in cloud computing in the work field.

8.0 References

Allen, S., 2017. *Developing with .NET on Microsoft Azure.* [Online] Available at: https://www.pluralsight.com/courses/developing-dotnet-microsoft-azure-getting-started [Accessed 30th October 2017].

Cecaro, F., 2015. *Azure Cloud Design Patterns.* [Online] Available at: https://cloudacademy.com/blog/azure-cloud-design-patterns/ [Accessed 28th October 2017].

Dwivedi, K., Wheeler, S., Madureira, J. & Tuliani, J., 2017. *Overview of Traffic Manager.* [Online] Available at: https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview [Accessed 7th November 2017].

Georgiev, G., 2014. *What is Vertical scaling and Horizontal scaling – Vertical and Horizontal hardware / services scaling.* [Online] Available at: http://www.pc-freak.net/blog/vertical-horizontal-server-services-scaling-vertical-horizontal-hardware-scaling/ [Accessed 5th November 2017].

Glover, G., 2015. *The Importance of Log Management.* [Online] Available at: http://blog.securitymetrics.com/2015/08/importance-of-log-management.html [Accessed 4th November 2017].

Lin, C., 2017. *Web Apps overview.* [Online] Available at: https://docs.microsoft.com/en-us/azure/app-service/app-service-web-overview [Accessed 1st November 2017].

Macken, T. F., Foulds, I., Stewart, R. & Squillace, R., 2017. *Azure Resource Manager overview*. [Online] Available at: https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-overview#the-benefits-of-using-resource-manager [Accessed 8th November 2017].

Microsoft, 2017. *Data Replication and Synchronization Guidance.* [Online] Available at: https://msdn.microsoft.com/en-us/library/dn589787.aspx [Accessed 29th October 2017].

Microsoft, 2017. *Traffic Manager routing methods.* [Online] Available at: https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-routing-methods [Accessed 7th November 2017].

Microsoft, 2017. *What is PaaS?.* [Online] Available at: https://azure.microsoft.com/en-us/overview/what-is-paas/ [Accessed 15th November 2017].

Narumoto, M. et al., 2017. *Autoscaling.* [Online] Available at: https://docs.microsoft.com/en-us/azure/architecture/best-practices/auto-scaling [Accessed 28th October 2017].

Rabeler, C. & Kerkhove, T., 2017. *Database Transaction Units (DTUs) and elastic Database Transaction Units (eDTUs).* [Online] Available at: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-what-is-a-dtu [Accessed 1st November 2017].

Rabeler, C. et al., 2017. *Choose a cloud SQL Server option: Azure SQL (PaaS) Database or SQL Server on Azure VMs (IaaS).* [Online] Available at: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-paas-vs-sql-server-iaas [Accessed 15th November 2017].

Sreeram, P. K., 2016. *Azure Scale up and Scale Out.* [Online] Available at: https://praveenkumarsreeram.com/2016/11/19/azure-scale-up-and-scale-out/ [Accessed 10th November 2017].

9.0 Appendices

Completed Lab Tutorials

Lab	Video URL
Lab 1 Building an Enterprise Web App with Azure and Visual Studio 2017	https://web.microsoftstream.com/video/3bfb1900-7d7a-43f9-9bfa-eed5309a7e7c
Lab 2 Build an Azure Web App with Visual Studio 2017	https://web.microsoftstream.com/video/68386510-2e77-4959-a54d-5d0f5ffe5791
Lab 3 Introduction to API Management	https://web.microsoftstream.com/video/b2b1a0b8-078d-4ea3-9a19-8370a5df8e04
Lab 4 Deploying a Secure Intranet App with Azure	https://web.microsoftstream.com/video/46f6f163-51b6-4a75-864f-cdd13ae9be5f
Lab 5 Creating & Migrating To Azure SQL Database	https://web.microsoftstream.com/video/04534ba5-4d5f-4e42-b451-e70c47def949

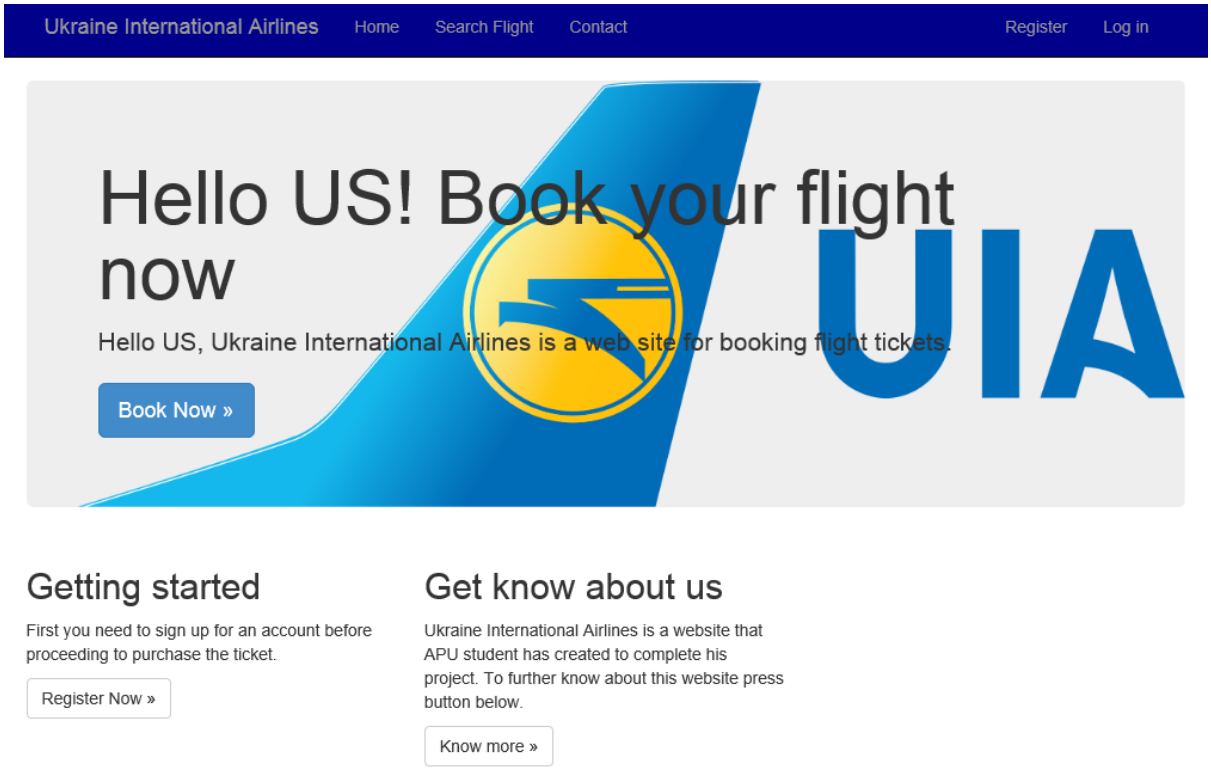
GitHub Repository

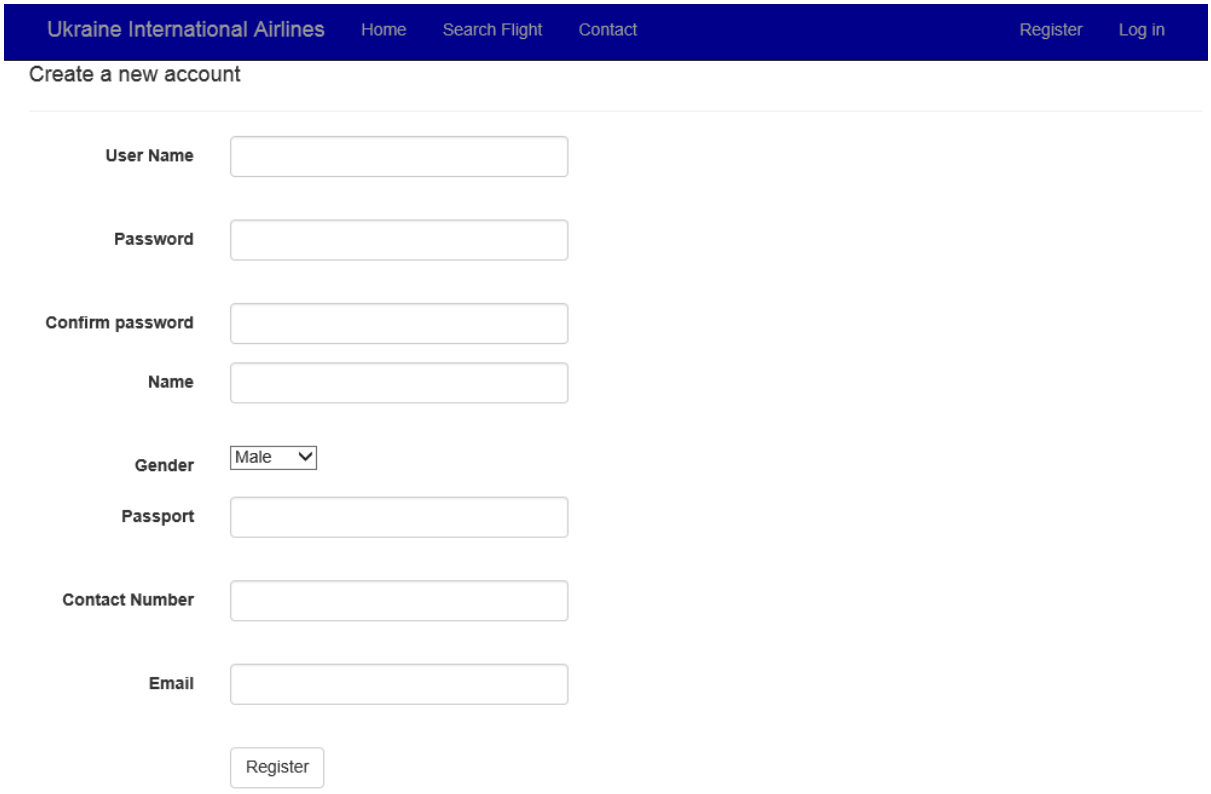
https://github.com/lamzh/TP035832UIA

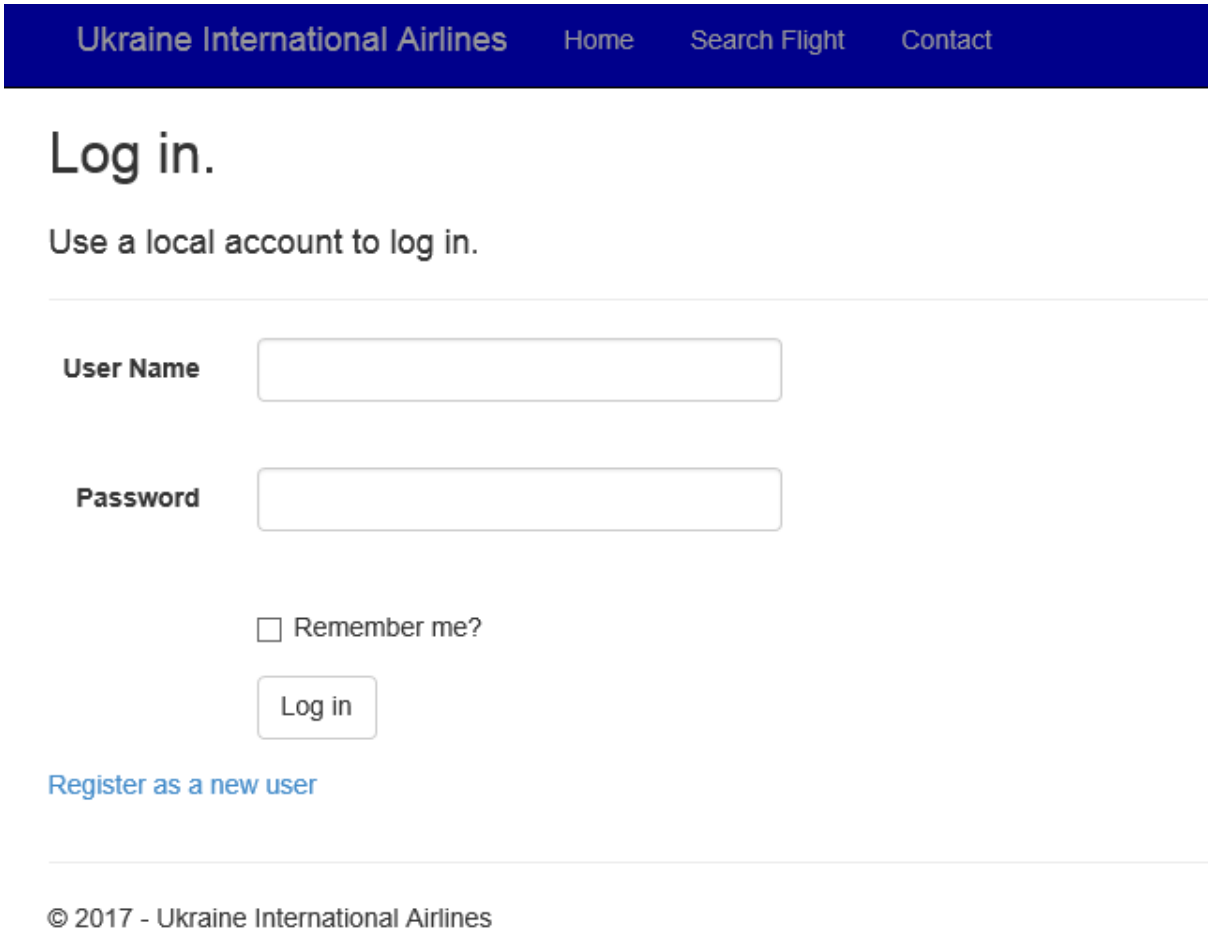
Link of Video Presentation

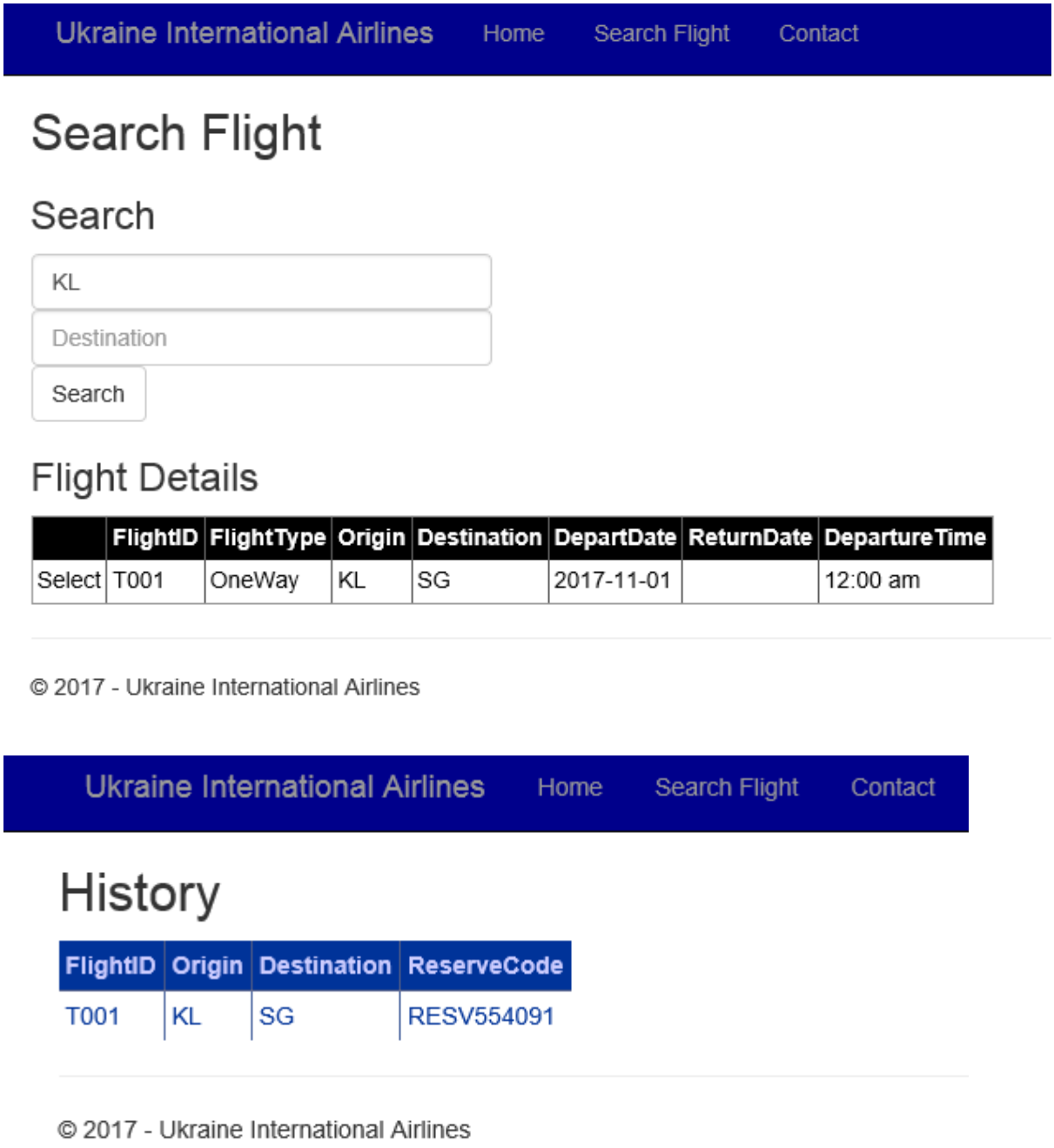
https://web.microsoftstream.com/video/432e7494-ff54-45fd-9370-0906bc2627b7

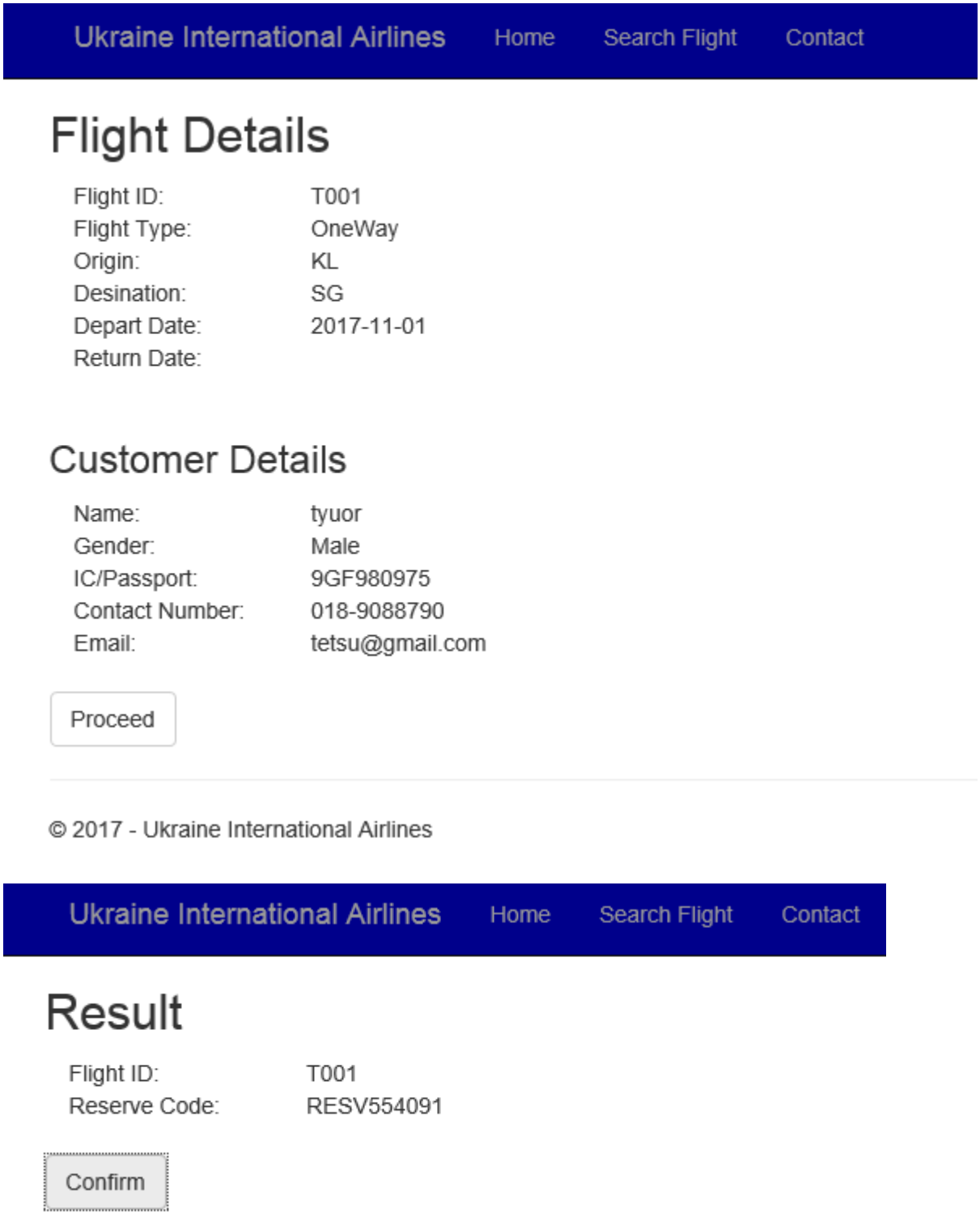
Interface of Web Application






Ukraine International Airlines
Home
Search Flight
Contact
Log in.
Use a local account to log in.
User Name
Password
Remember me?
Log in
Register as a new user
© 2017 - Ukraine International Airlines


Ukraine International Airlines
Home
Search Flight
Contact
Search Flight
Search
KL
Destination
Search
Flight Details
FlightID FlightType Origin Destination DepartDate ReturnDate DepartureTime
Select T001 OneWay KL SG 2017-11-01 12:00 am
© 2017 - Ukraine International Airlines
Ukraine International Airlines
Home
Search Flight
Contact
History
FlightID Origin Destination ReserveCode
T001 KL SG RESV554091
© 2017 - Ukraine International Airlines


Ukraine International Airlines
Home
Search Flight
Contact
Flight Details
Flight ID: T001
Flight Type: OneWay
Origin: KL
Desination: SG
Depart Date: 2017-11-01
Return Date:
Customer Details
Name: tyuor
Gender: Male
IC/Passport: 9GF980975
Contact Number: 018-9088790
Email: tetsu@gmail.com
Proceed
© 2017 - Ukraine International Airlines

Ukraine International Airlines
Home
Search Flight
Contact
Result
Flight ID: T001
Reserve Code: RESV554091
Confirm