Acknowledgement

Firstly, I would like express my gratitude to Dr. Kalai for his guidance in the field of cloud computing which I have gain knowledge for future use. I also would like to thank my friends for helping me by providing opinions so that I could develop and deploy the project on the right track.

Contents

[1.0 Introduction 3](#_Toc495332717)

[1.1 Project Background 3](#_Toc495332718)

[1.2 Project Objective 3](#_Toc495332719)

[1.3 Project Scope 3](#_Toc495332720)

[1.4 Project Deliverables 3](#_Toc495332721)

[2.0 Project Plan 4](#_Toc495332722)

[2.1 Gantt Chart 4](#_Toc495332723)

[3.0 Design 5](#_Toc495332724)

[3.1 Modelling 5](#_Toc495332725)

[3.1.1 Use case 5](#_Toc495332726)

[3.1.2 Data Modelling 6](#_Toc495332727)

[3.1.3 Sequence Diagram 7](#_Toc495332728)

[3.1.4 Site Map 10](#_Toc495332729)

[3.2 Cloud Architecture 11](#_Toc495332730)

[4.0 Implementation 12](#_Toc495332731)

[4.1 Application Delovelopment 12](#_Toc495332732)

[4.1.1 Screenshots and sample codes 13](#_Toc495332733)

[4.2 Deployment 18](#_Toc495332734)

[4.3 Application Scaling 19](#_Toc495332735)

[4.3.1 Traffic Manager 19](#_Toc495332736)

[4.3.2 Auto Scale 20](#_Toc495332737)

[4.4 Testing 21](#_Toc495332738)

[4.4.1 Unit Testing 21](#_Toc495332739)

[4.4.2 Performance Test 23](#_Toc495332740)

[4.5 Managed Database 24](#_Toc495332741)

[4.5.1 Common PaaS scenarios 25](#_Toc495332742)

[4.5.2 Advantages 26](#_Toc495332743)

[5.0 Conclusion 27](#_Toc495332744)

[6.0 Reference 28](#_Toc495332745)

# 1.0 Introduction

## 1.1 Project Background

Ukraine International Airlines (UIA), is looking at designing and developing an Online Flight Booking System. UIA looked at both Microsoft Azure and Amazon Web Services and chose Azure. Thus, the UIA Flight Booking System will be developed and deployed in Microsoft Azure platform.

## 1.2 Project Objective

- To develop the online flight booking system and deploy on Azure in highest performance.

## 1.3 Project Scope

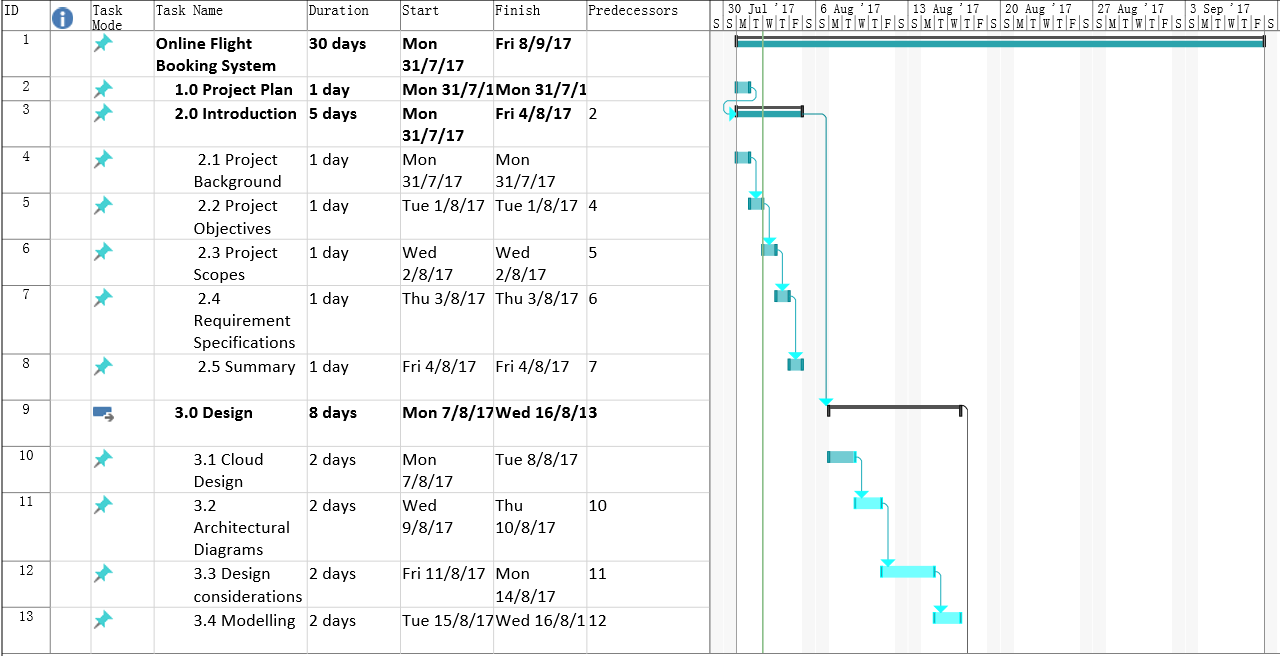
The system will be designed with modelling, the will undergo application development. Once the development is done, it will be tested and deployed in Microsoft Azure. Configuration in Azure is such as traffic manager will also be implemented and performance test is also conducted

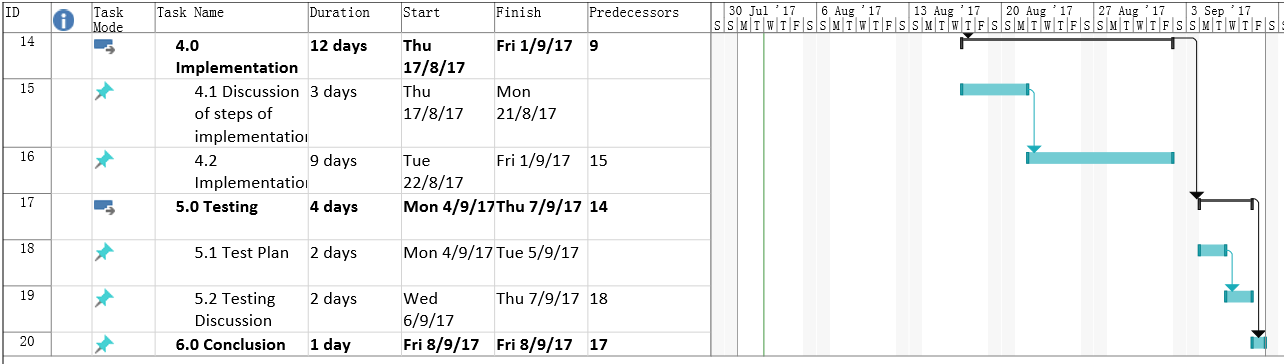
## 1.4 Project Deliverables

* To allow users to register
* To allow users to login
* To allow users to search flight
* To allow users to book flight
* To allow users to view booking

# 2.0 Project Plan

## 2.1 Gantt Chart

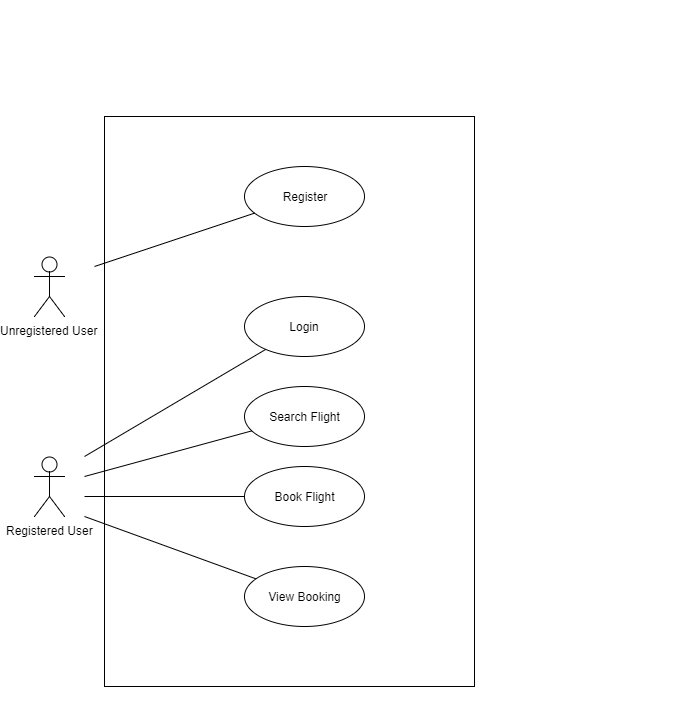




# 3.0 Design

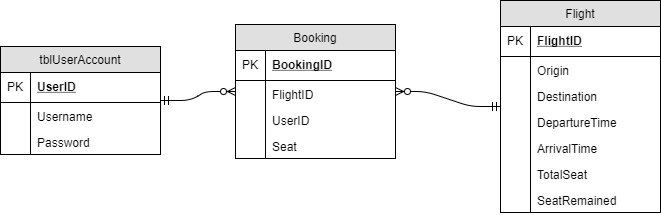
## 3.1 Modelling

### 3.1.1 Use case



Use case diagram is used to describe the functionality of the system. Unregistered users require to register an account perform functionality. Registered users are able to search flight, book flight, and view booking.

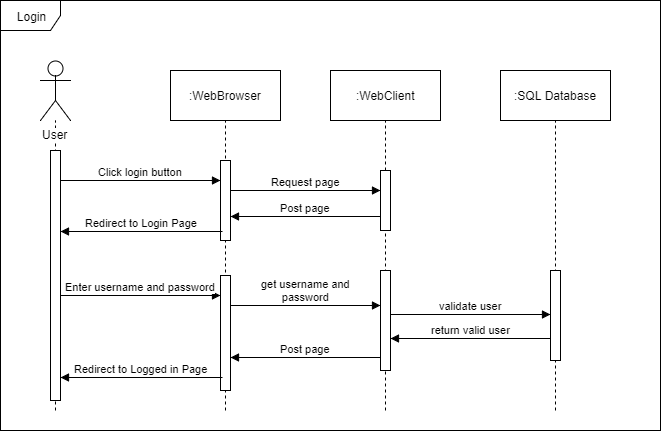
### 3.1.2 Data Modelling



The ERD shows the tables for the database. The tblUserAccount table has all the information for the user. The flight table stores the details of flight and booking table will store the user’s information, flight’s information and the seat number booked.

### 3.1.3 Sequence Diagram

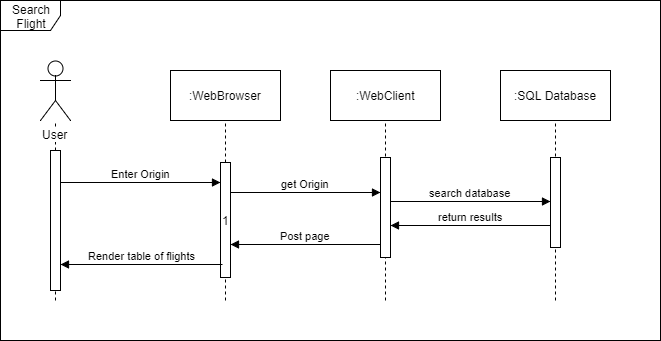
Login



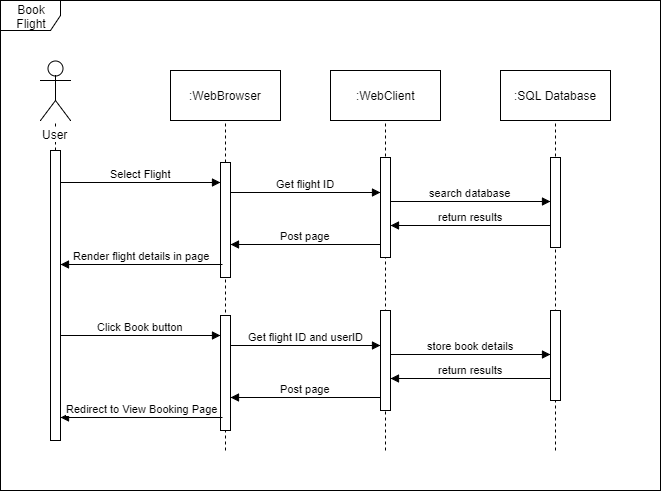
Registration



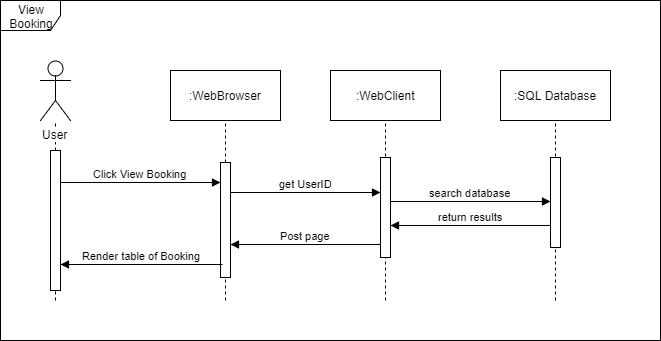
Search flight



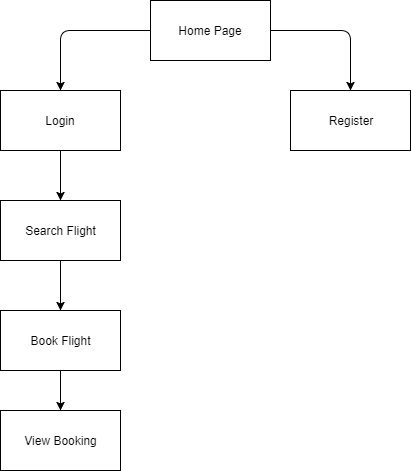
Book Flight



View Booking

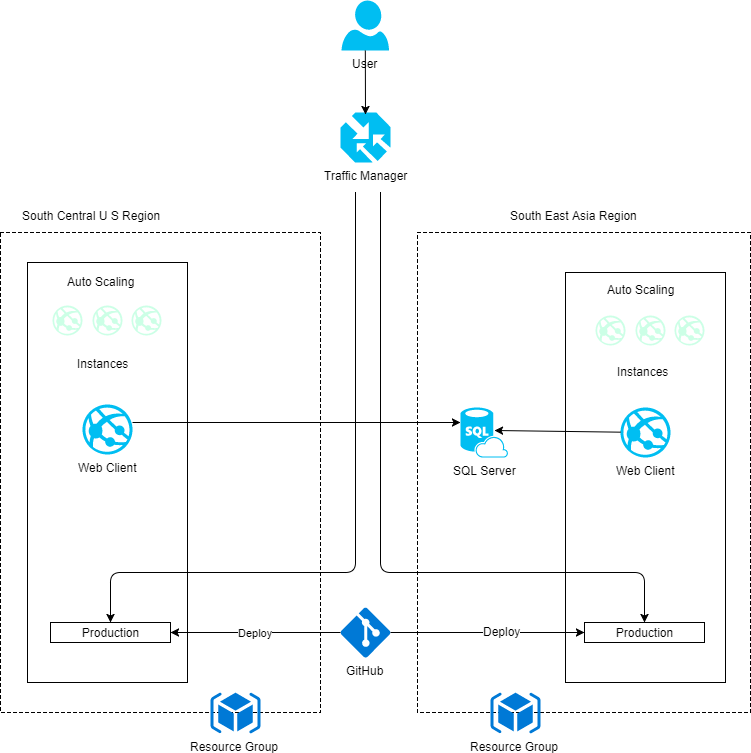


### 3.1.4 Site Map



Site Map shows the flow of the functionality of the system.

## 3.2 Cloud Architecture

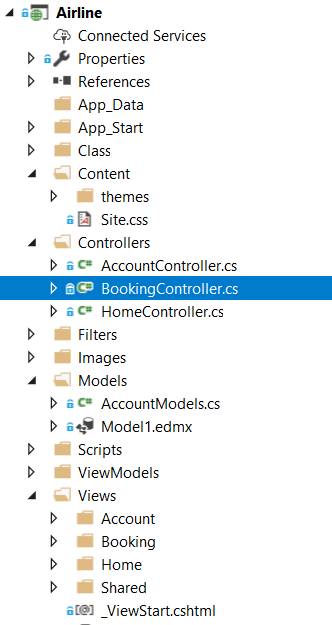


The diagram shows the cloud architecture for deploying the application into Azure platform. The Flight booking system will be deployed in two different regions which is South East Asia Region and South Central US region. The application is also managed by traffic manager to ensure high performance of the application. Local SQL Server is also migrated to Azure SQL Database. The source code of the application will be uploaded to GitHub and through GitHub, the application will deploy to Azure website.

# 4.0 Implementation

The source code of the application is available at <https://github.com/wyk815/DDAC-UIAirline>

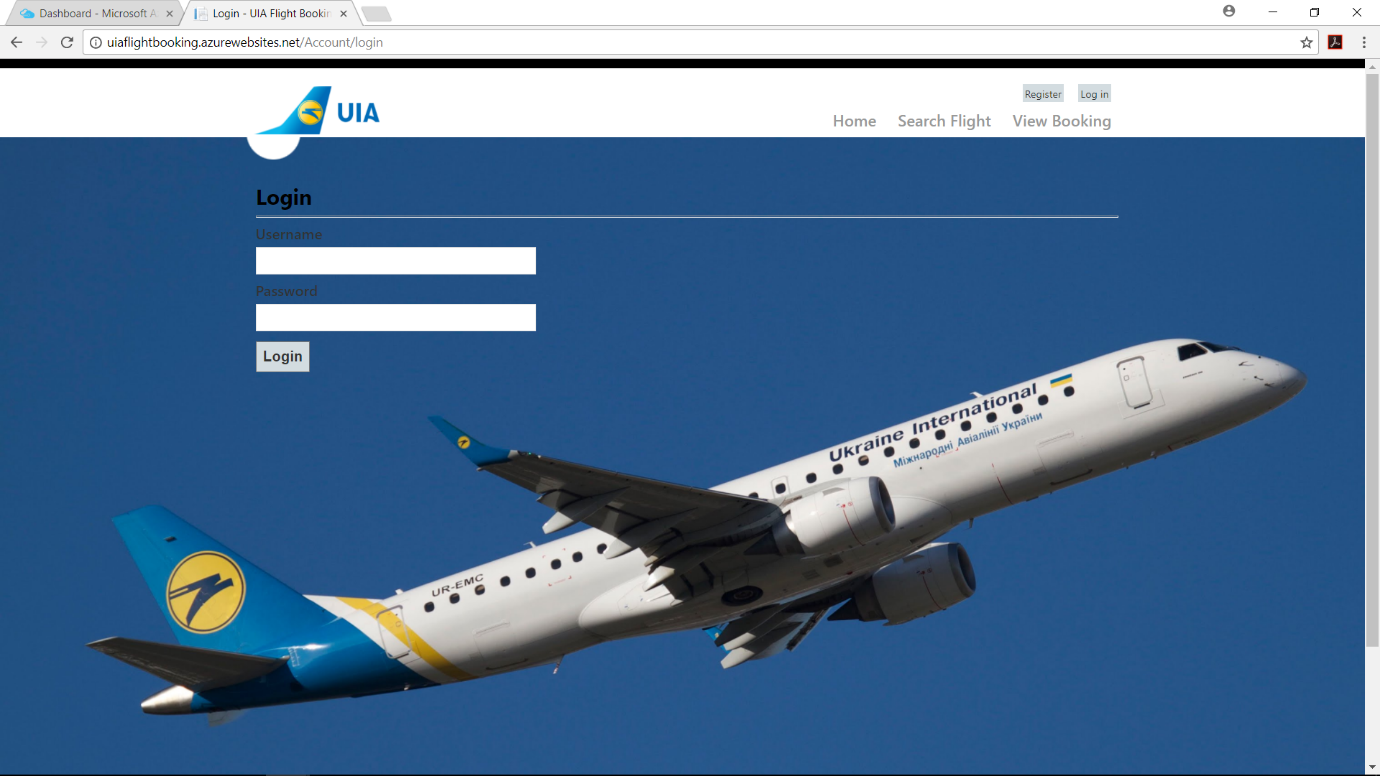
## 4.1 Application Delovelopment

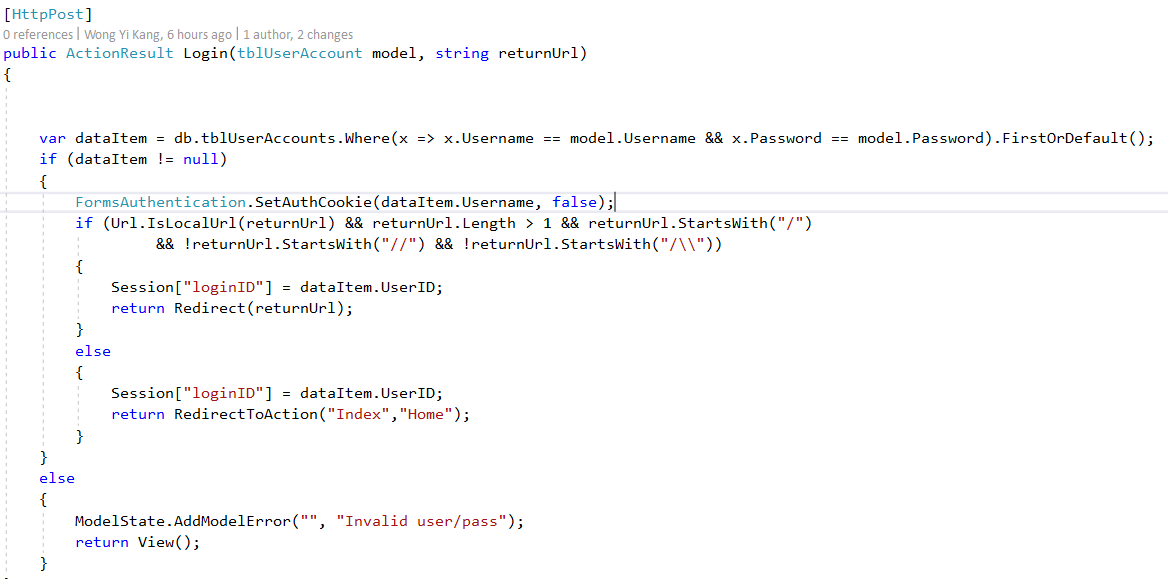


The system was developed using C# MVC and Sql Server Management Studio as database. The structure consists of model, view and controllers. Model acts as medium which the data are being transferred between the controllers and views. Views are the user interface components and controllers act as an interface between model and view components to process the logic, and incoming request.

### 4.1.1 Screenshots and sample codes

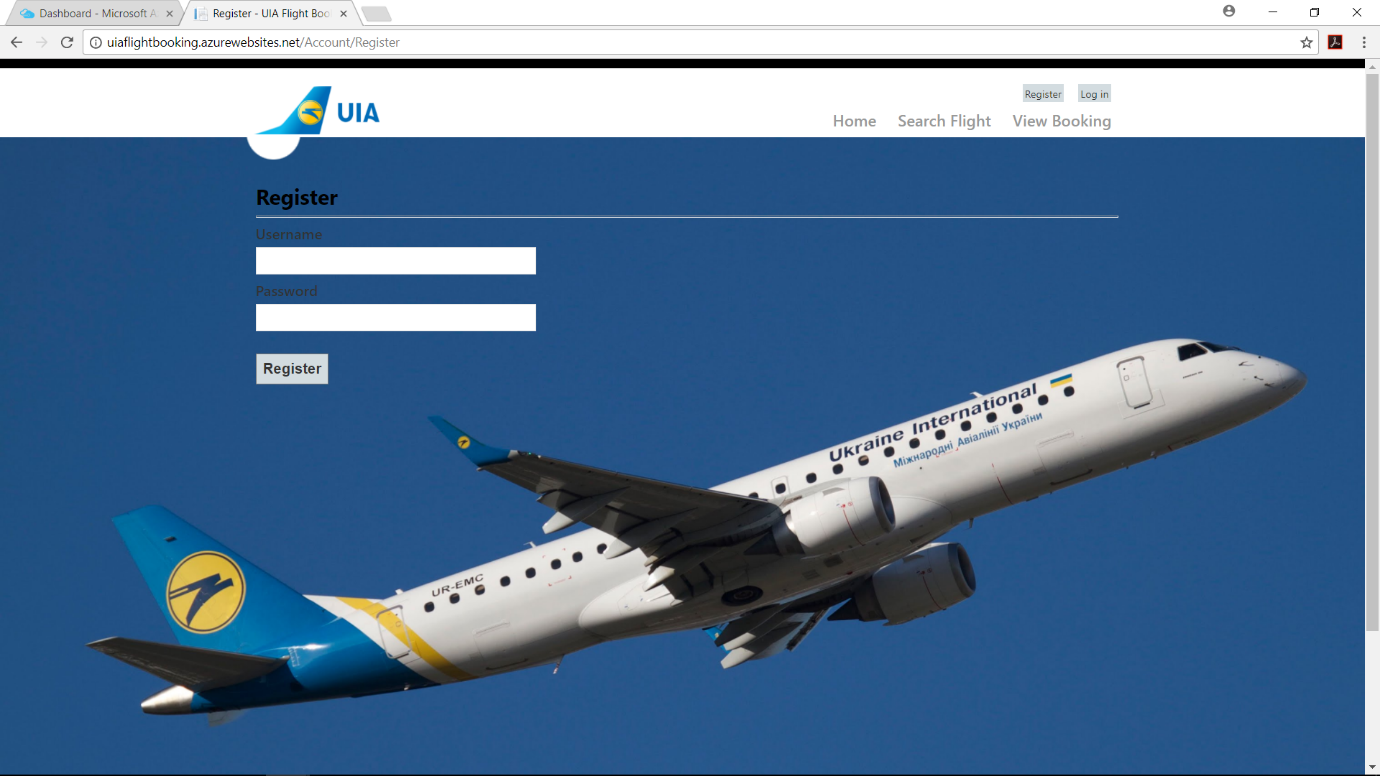
**Login**

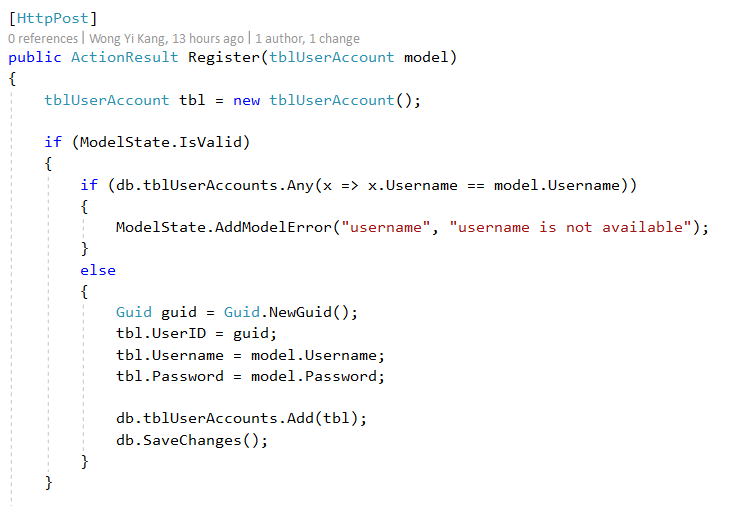




The code above shows the login implement for the system. The username and password entered will be retrieved from view to controllers and query of linQ is used to validate the account. If valid the username will be set in cookie for identification use. userID is also stored in session as well.

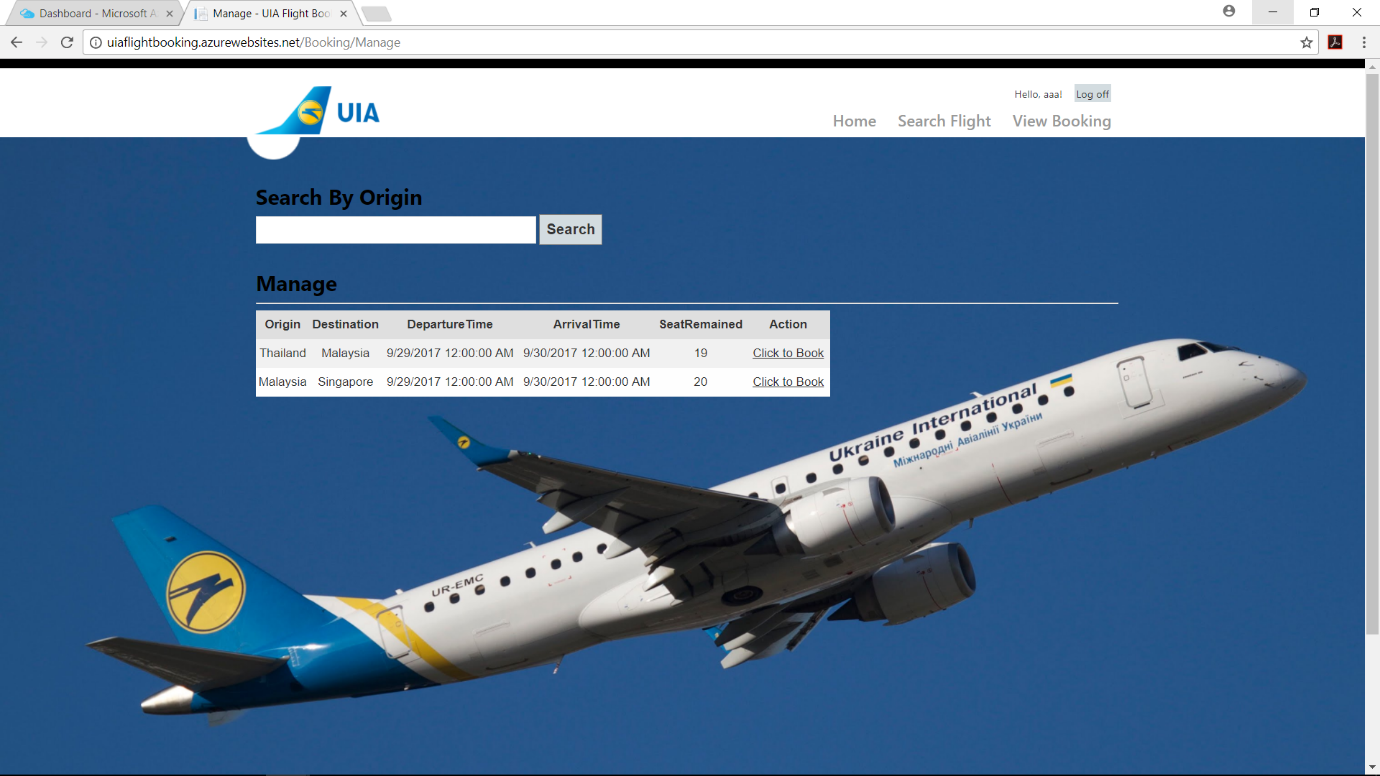
**Registration**

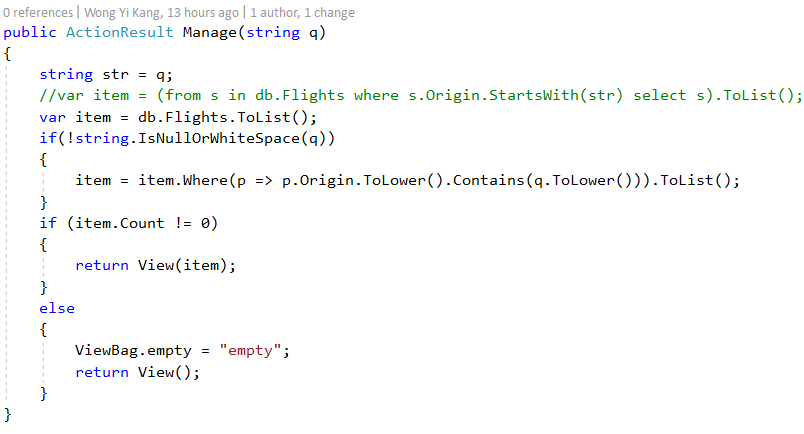




The diagram shows registration for the system. The username will be checked for duplication.

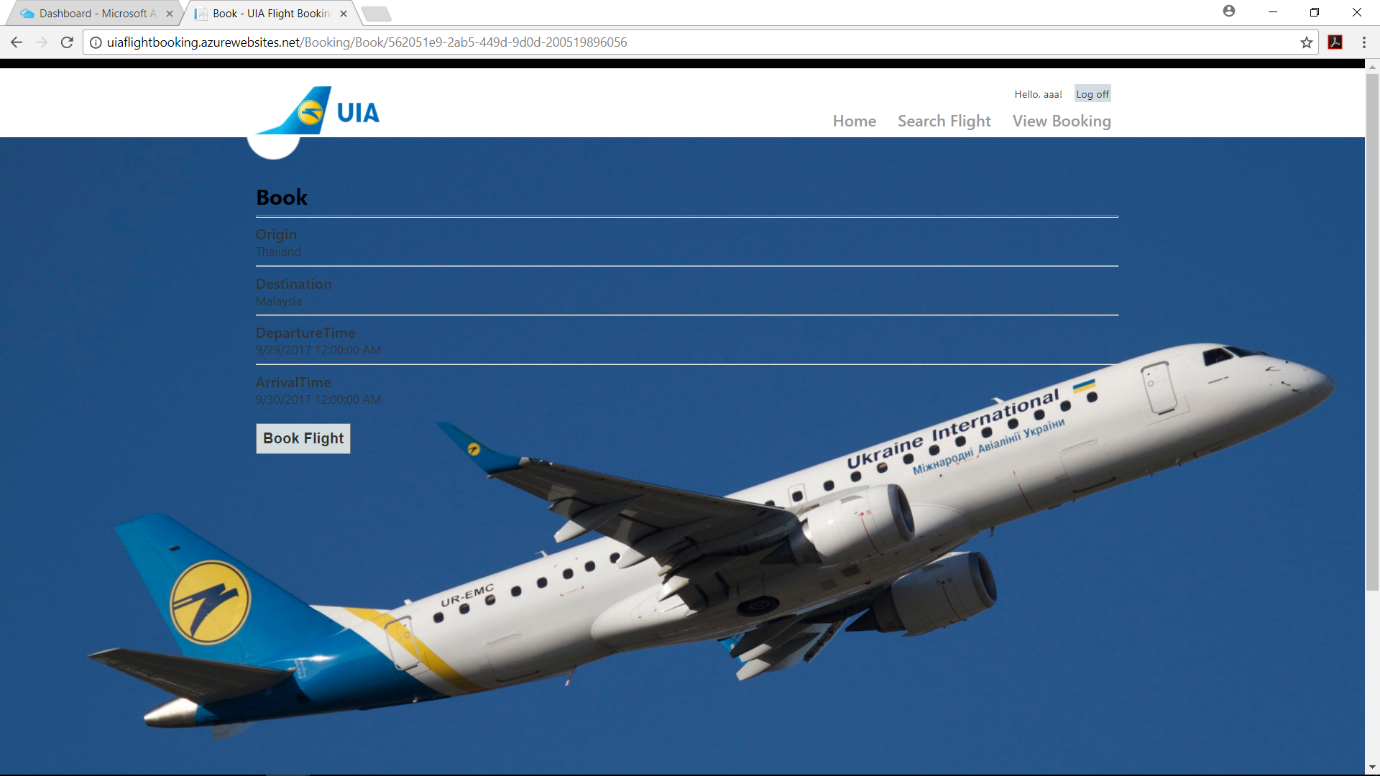
**Search Flight**

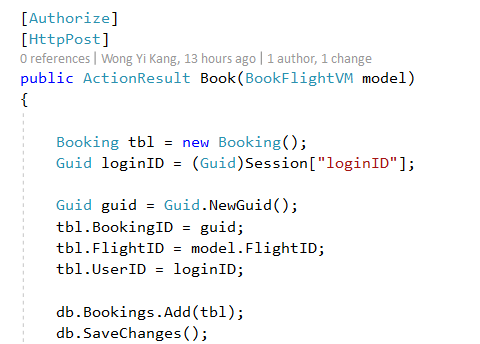




The search functionality will be search by origin. String isNullOrWhiteSpace is used to check the user input is null or white space. If it is not then the linQ will query to search for origin entered with case insensitive implemented.

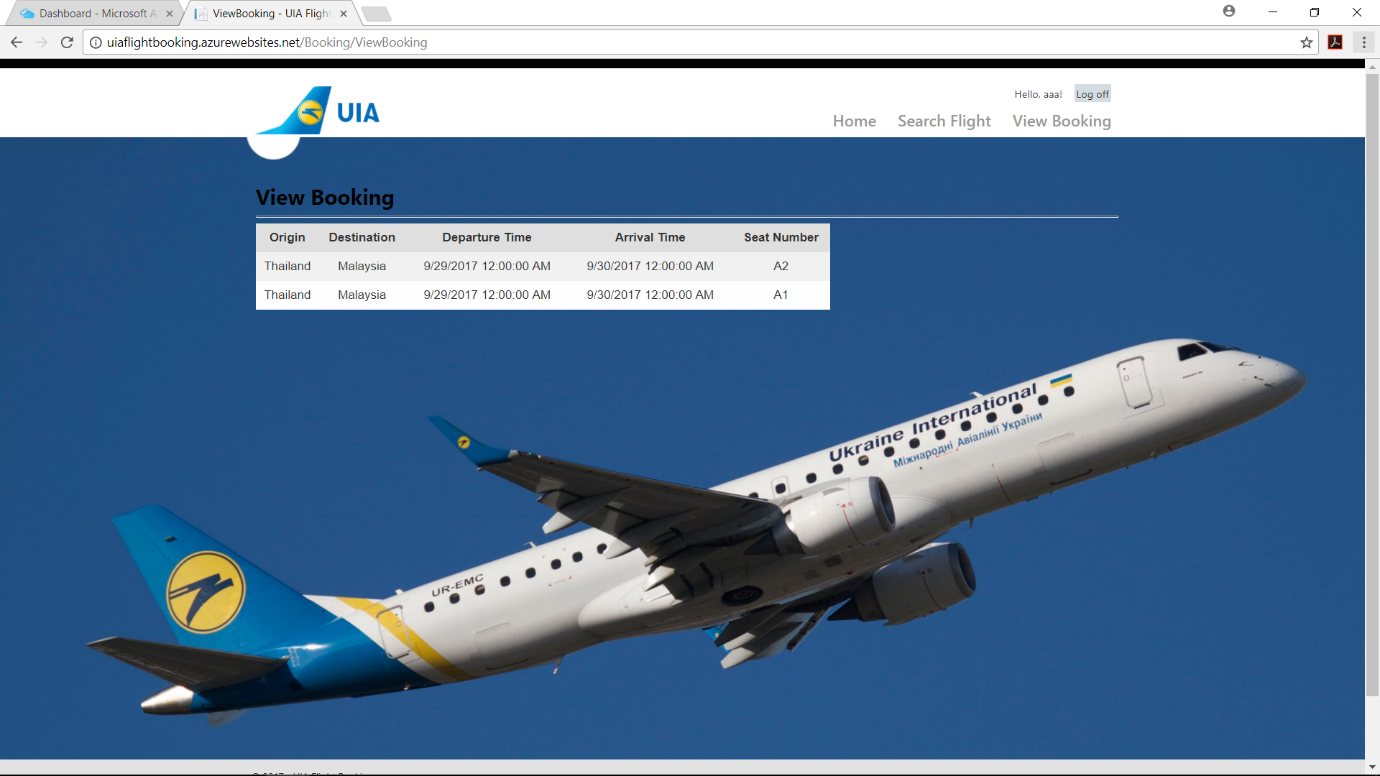
**Book Flight**

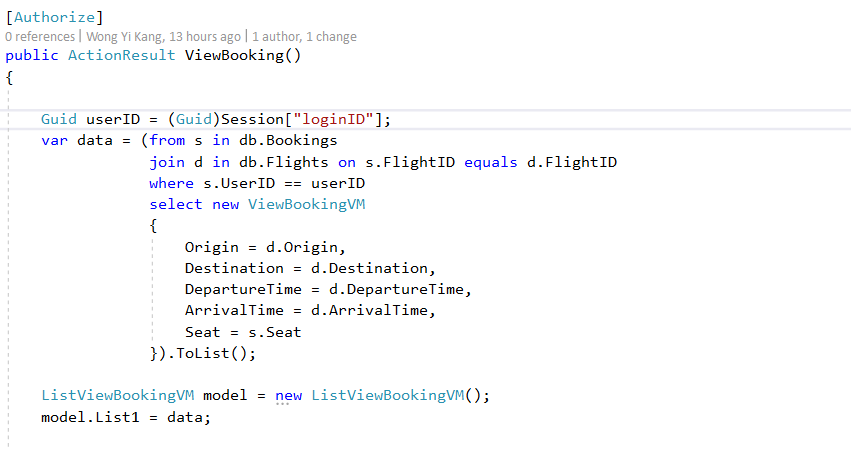




The diagram shows the booking for the system. The details of flight as well as user’s details will be added to booking table.

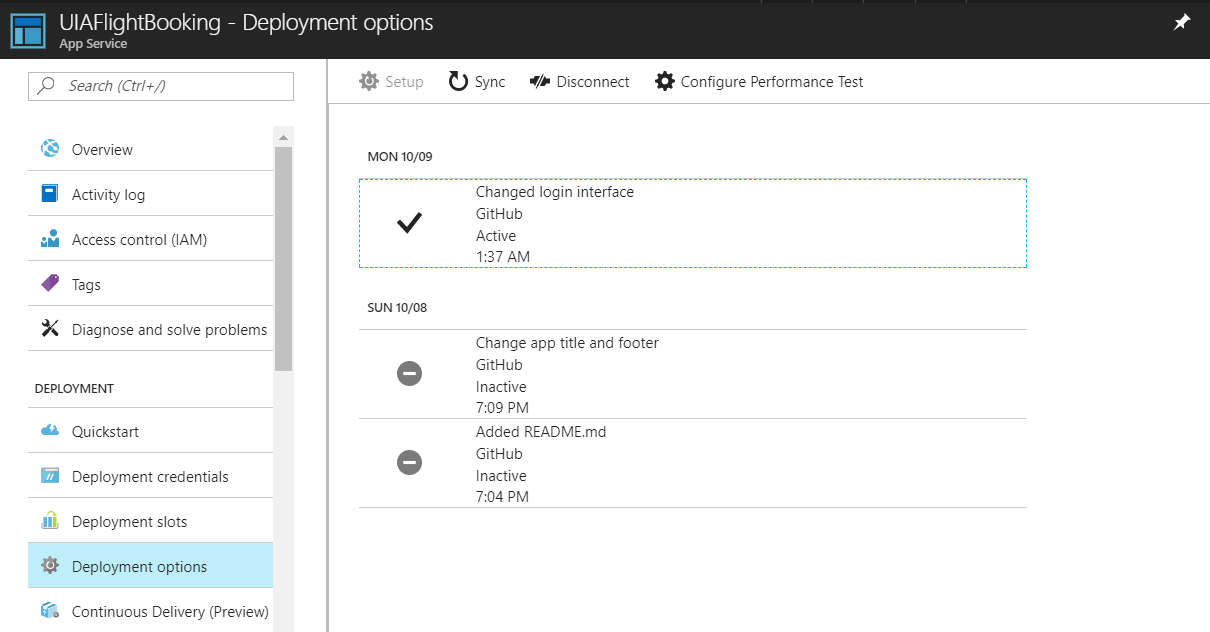
**View Booking**





The booking can be view through this code. The inner join is implement to view flight details in flight table as well as booking details in booking table.

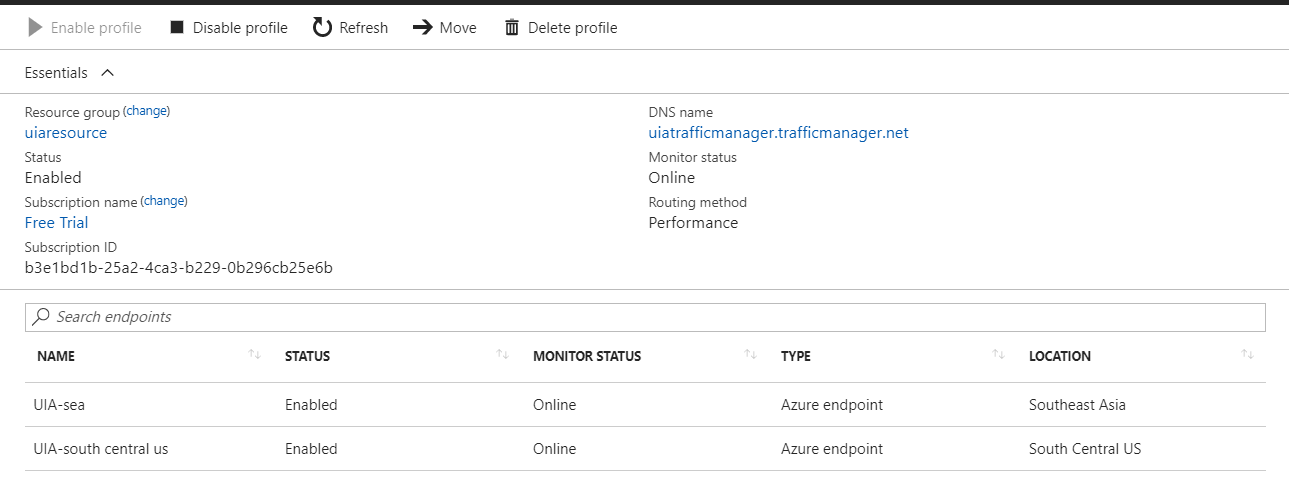
## 4.2 Deployment



The system is deployed using Azure automating deployment options through GitHub. This enables changes in development can be easily pushed to a particular branch in GitHub, and then the GitHub will automatically deploy to Azure. The changes made are shows in the diagram above.

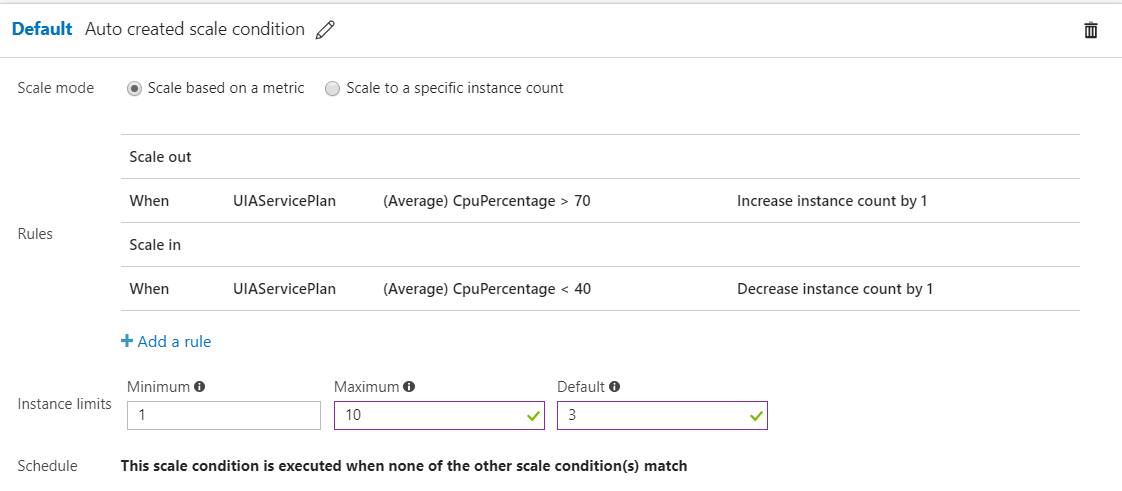
## 4.3 Application Scaling

### 4.3.1 Traffic Manager



Microsoft Azure Traffic Manager allows to control the distribution of user traffic for service endpoints in different datacenters. It allows the website to improve the responsiveness of the system by redirecting the traffic to the endpoint with lowest network latency for user. For example, users in Southeast Asia will redirect to SEA endpoint, reducing the time of response of the system thus improving the performance of the system

### 4.3.2 Auto Scale





Autoscale is a built-in feature of Cloud Services, Mobile Services, Virtual Machines, and Websites that helps applications perform their best when demand changes. (Microsoft, 2017) The performance means different things for different applications. Some apps are CPU-bound, others memory-bound. For example, you could have a web app that handles millions of requests during the day and none at night. Autoscale can scale your service by any of these—or by a custom metric you define. Each region’s app service will run 3 instances of up to 10 instance for higher reliability. Moreover, it will increase the instance by 1 when the CPU percentage is more than 70%. The instance will decrease by 1 when CPU percentage is less 40%.

## 4.4 Testing

### 4.4.1 Unit Testing

Login Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Test Name | Test Description | Expected Results | Actual Results |
| L1 | Invalid user | Entered wrong username or password | Return Invalid user/pass message in login page | Expected |
| L2 | Valid user | Enter valid username and password | Redirect to home page | Expected |

Registration Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Test Name | Test Description | Expected Results | Actual Results |
| R1 | Username Duplication | Enter existing username in registration page | Return username is not available message | Expected |
| R2 | Registration | Enter username and password | Redirect to login page | Expected |

Search Flight

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Test Name | Test Description | Expected Results | Actual Results |
| S1 | Search | Enter valid search by origin | Return results of flight table | Expected |
| S2 | Case insensitive | Enter search by origin with lower case | Return results of table with origin of upper case letter as well | Expected |
| S3 | Not Found | Enter invalid search by origin of flight | Return no result message | Expected |

Book Flight

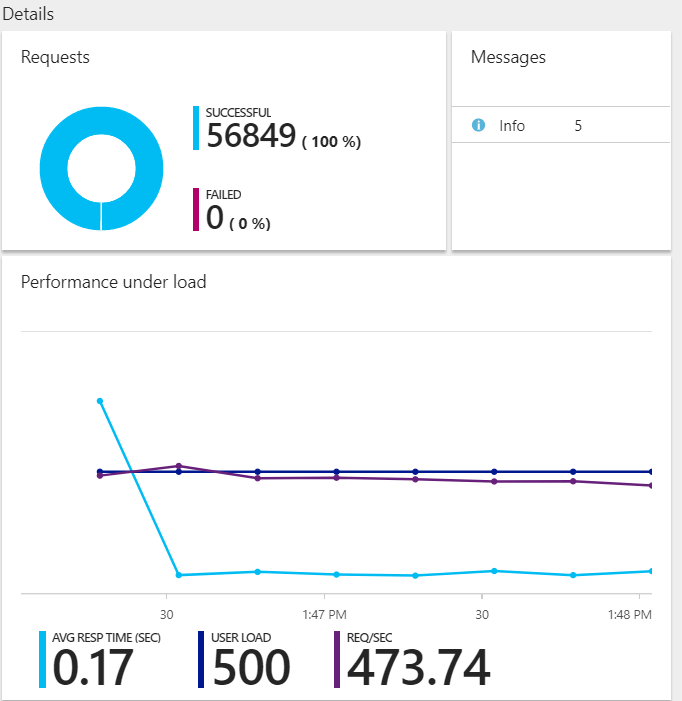
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Test Name | Test Description | Expected Results | Actual Results |
| B1 | Click to Book | Press href link of Click to Book in flight table | Redirect to booking page with flight details displayed | Expected |
| B2 | Book | Click Book Button to book | Book success | Expected |

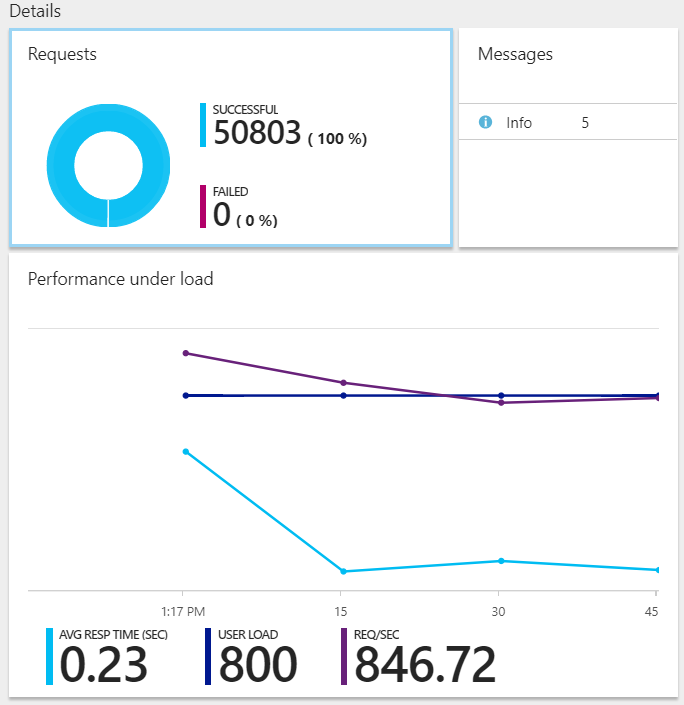
View Flight

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Test Name | Test Description | Expected Results | Actual Results |
| V1 | View Flight | Click view booking button in home page | Redirect to view booking page with booking details displayed | Expected |

### 4.4.2 Performance Test

The performance test is conducted in Azure app service. It will be tested start with 300 with the increment of 200 users until 800 users within 2 minutes duration time. The diagram below shows the average response time for certain amount of concurrent users.

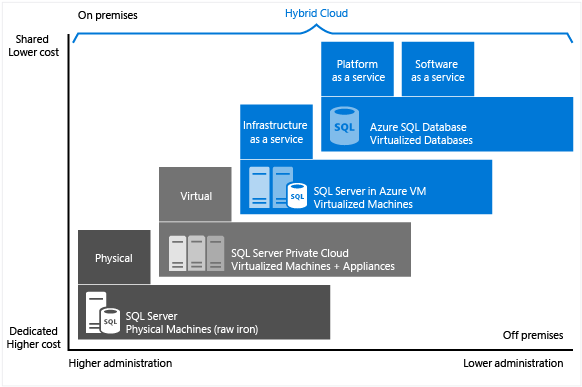




|  |  |  |  |
| --- | --- | --- | --- |
| Concurrent User | 300 | 500 | 800 |
| Average Response Time (sec) | 0.1 | 0.17 | 0.23 |

## 4.5 Managed Database

A SQL database native to the cloud, also known as a platform as a service (PaaS) database or a database as a service (DBaaS) that is optimized for software-as-a-service (SaaS) app development. It offers compatibility with most SQL Server features.



As it can be seen in the diagram above provided by Microsoft, the cost and the administration gets lower transferring from physical SQL server to Azure SQL Database (PaaS) and yet it still enables advanced features. It supports development team that need built-in high availability as well as disaster recovery for their database.

### 4.5.1 Common PaaS scenarios

Development framework

PaaS provides a framework that developers can build upon to develop or customize cloud-based applications. Similar to the way you create an Excel macro, PaaS lets developers create applications using built-in software components. Cloud features such as scalability, high-availability, and multi-tenant capability are included, reducing the amount of coding that developers must do.

Analytics or business intelligence

PaaS provides Tools as services to allow organizations to analyse and mine their data, finding insights and patterns and predicting outcomes to improve forecasting, product design decisions, investment returns, and other business decisions.

Additional services

PaaS providers may offer other services that enhance applications, such as workflow, directory, security, and scheduling.

### 4.5.2 Advantages

**Low risk and initial investment**

The use of the software without having to make an initial investment in machinery, software base (so) and additional software to run the application they would need in an environment in-house, is an important benefit for IT managers and ultimately for the company.

**Cost reduction**

In addition to paying for only what you need, you get a cost savings of maintaining the platform of machinery and the necessary software (databases, application server) to run your application.

**Immediate updates and new features**

A party that will not require dedicated staff for updates and software upgrades immediately. Sometimes we choose to use new features.

**Support more agile and quick implementation**

Bugs have a direct treatment and commissioning solution faster than in-house facilities. It enables faster time to market by allowing development teams to focus on the application

**Increased availability and data security**

Provides features such as automated backups, point-in-time restore, geo-restore.

# 5.0 Conclusion

In conclusion, the development of the application is indeed challenging as well as the deployment onto Azure. Deep understanding in utilizing the cloud services and setting up the application must be considered in order to maintain the performance of application. For example, traffic manager and auto scaling is implemented to control as well as maintain the request from users to improve the response time of the application thus improving the application performance. Furthermore performance test is conducted in order to ensure that the application is able to run in heavy or certain numbers of concurrent users without severely affecting the performance of the web application.

# 6.0 Reference

Microsft. 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 9 October 2017].

Judith Hurwitz, Marcia Kaufman, Fern Halper, Daniel Kirsch. 2017. THE BUSINESS BENEFITS OF PAAS IN CLOUD COMPUTING. [ONLINE] Available at: http://www.dummies.com/programming/cloud-computing/hybrid-cloud/the-business-benefits-of-paas-in-cloud-computing/. [Accessed 9 October 2017].

Pravin Ganore. 2011. Advantages and disadvantages of SaaS and PaaS. [ONLINE] Available at: https://www.esds.co.in/blog/advantages-and-disadvantages-of-saas-and-paas/#sthash.DRDPQu3r.dpbs. [Accessed 9 October 2017].

Microsoft. 2017. Manage an Azure Traffic Manager profile. [ONLINE] Available at: https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-manage-profiles. [Accessed 9 October 2017].

Microsoft. 2017. Autoscaling. [ONLINE] Available at: https://docs.microsoft.com/en-us/azure/architecture/best-practices/auto-scaling. [Accessed 9 October 2017].

Microsoft. 2017. Best practices for Autoscale. [ONLINE] Available at: https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/insights-autoscale-best-practices. [Accessed 9 October 2017].

Microsoft. 2017. How to configure auto scaling for a Cloud Service in the classic portal. [ONLINE] Available at: https://docs.microsoft.com/en-us/azure/cloud-services/cloud-services-how-to-scale. [Accessed 9 October 2017].