

Individual Assignment  
CT071-3-3-DDAC  
Designing & Developing Applications on the Cloud

Name: Lin Siong Jie

TP Number: TP032265

Assignment Title: Ukraine International Airlines (UIA)

Lecturer Name: DR. KALAI ANAND A/L RATNAM

Due Date: 20th November 2017

# 1.0 Acknowledgement

Firstly, I would like to thank my parents for their continuous support in my studies I would also like to thank my lecturer Dr Kalai for his guidance and expertise in the field of cloud computing. I would also like to thank my friends and fellow classmates for providing feedback to my ideas that would allow me to maintain the right path. My gratitude is also extended to Elon Musk for being an inspiration for me in continuing my journey into software engineering. I would also like to thank the Wiremod community for Garry’s mod for giving me an early understanding into the world of software and computing.

Table of Contents

[1.0 Acknowledgement 2](#_Toc498945149)

[2.0 Introduction 4](#_Toc498945150)

[3.0 Project Plan 5](#_Toc498945151)

[4.0 Design 6](#_Toc498945152)

[4.1 Design Consideration 6](#_Toc498945153)

[4.2 Modelling 6](#_Toc498945154)

[4.2.1 Use Case Diagram 6](#_Toc498945155)

[4.2.2 Sequence Diagram 7](#_Toc498945156)

[4.2.3 Entity Relationship Diagram 10](#_Toc498945157)

[4.2.4 Site Map 11](#_Toc498945158)

[4.3 Cloud Architecture 12](#_Toc498945159)

[5.0 Implementation 13](#_Toc498945160)

[5.1 Application Development 13](#_Toc498945161)

[5.2 Azure Publishing 13](#_Toc498945162)

[5.3 Application Scaling 15](#_Toc498945163)

[6.0 Test Plan 17](#_Toc498945164)

[6.1 Unit Testing 17](#_Toc498945165)

[6.2 Performance Testing 21](#_Toc498945166)

[7.0 Manage Databases 22](#_Toc498945167)

[8.0 Conclusion 23](#_Toc498945168)

[9.0 References 24](#_Toc498945169)

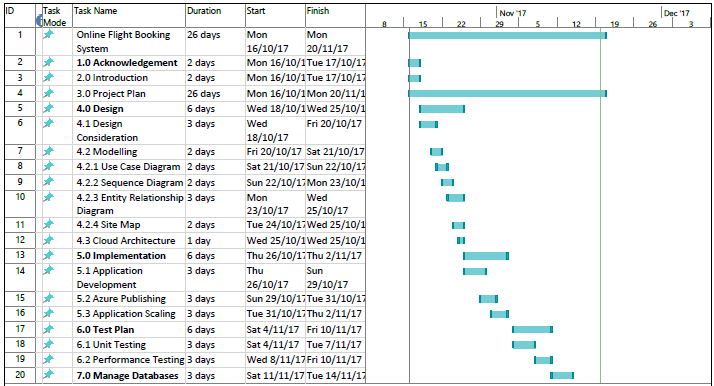
# 2.0 Introduction

Ukraine International Airlines (UIA) is the flagship carrier and largest airline in Ukraine. It operates domestic and international passenger flights and cargo services to Europe, the Middle East, the United States, and Asia. 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. the developer proposed the development of UIA Online Flight Booking System and deployment onto the Azure cloud platform.

The application provides UIA’s clients the ability to create accounts and book flights online. Because UIA is looking to expand, the need for high availability and performance of said services has become crucial to the company’s success. This document details the development of the solution to UIA’s problems as well as how the resulting system is deployed onto the Azure cloud. It will also explain the decisions made in the designing and development of the system.

This report covers the development of the proposed solution, thus providing detailed explanation on how the application system will be deployed onto the Azure cloud and the testing of the application system for reliability and performance in the Azure cloud.

# 3.0 Project Plan



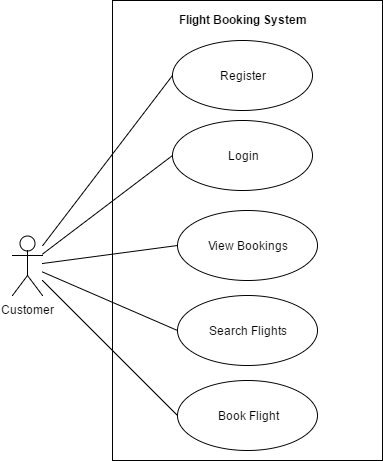
# 4.0 Design

## 4.1 Design Consideration

To design the application on Azure cloud platform, there are some considerations made by UI Airlines. One of which is the team have been provided with remunerations of Azure credits up to RM150 per month, as the team is required to build the new application system to proof that it is better than the current system. UI Airlines also plans on emphasizing the application system on the US and Southeast Asia market, therefore load balancing considerations are recommended.

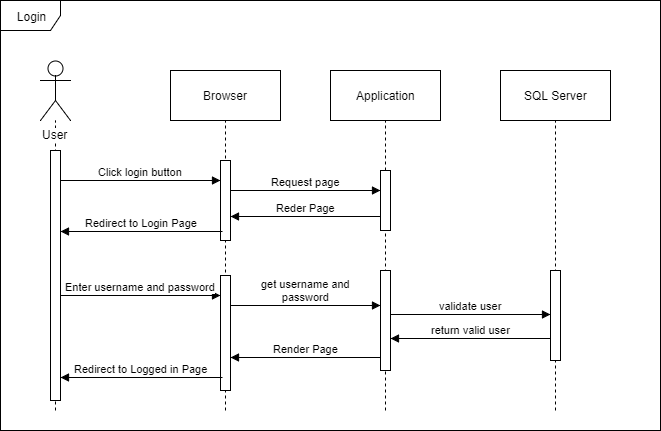
## 4.2 Modelling

### 4.2.1 Use Case Diagram

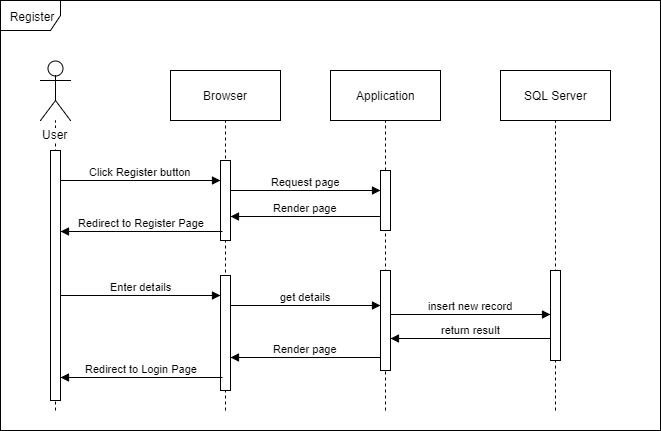


### 4.2.2 Sequence Diagram

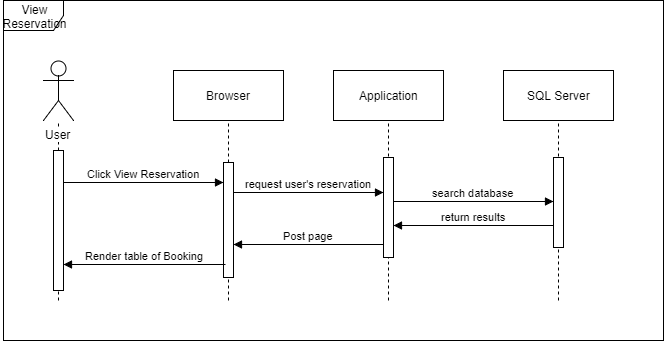
Login



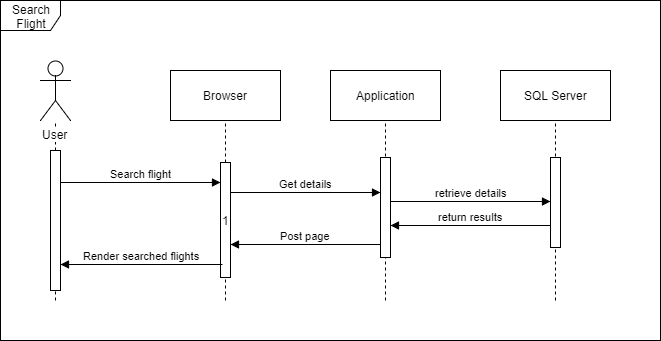
Register



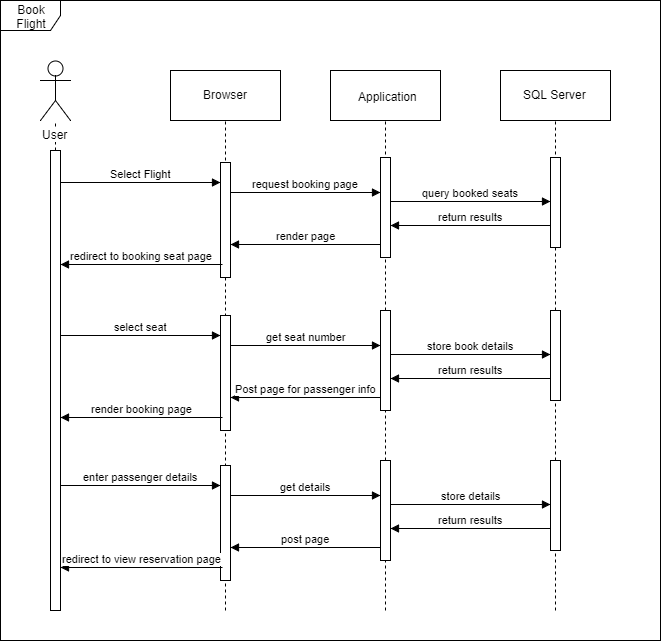
View Reservation



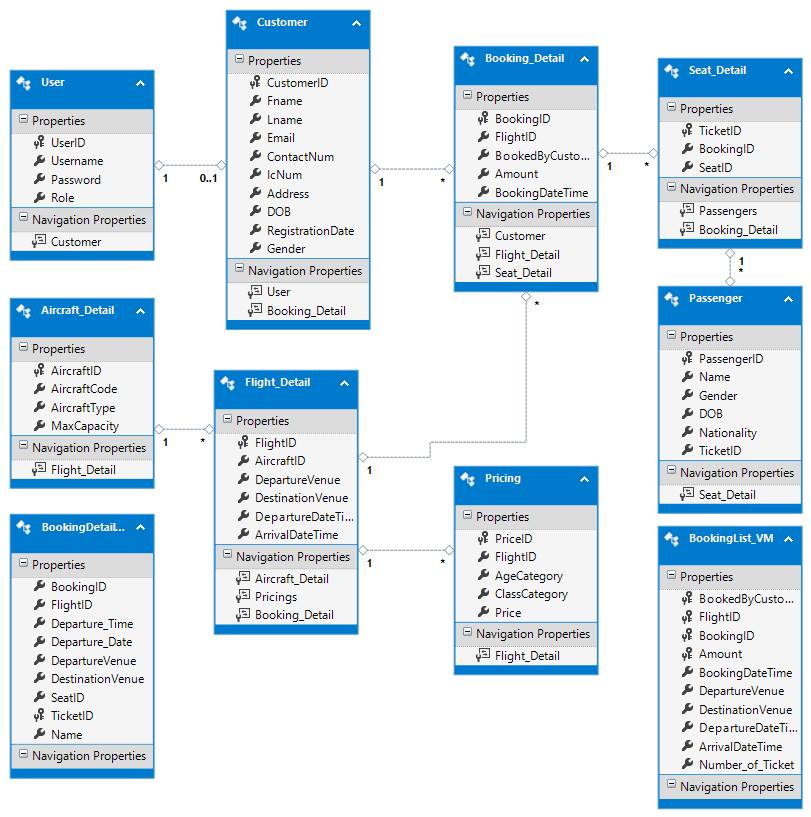
Search Flight



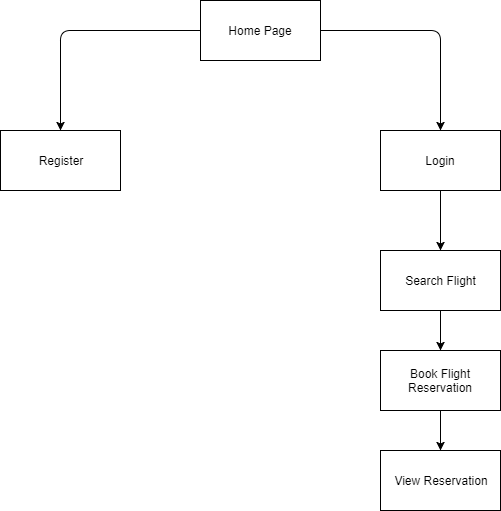
Book Flight Reservation



### 4.2.3 Entity Relationship Diagram



### 4.2.4 Site Map



## 4.3 Cloud Architecture

A close up of text on a white background

Description generated with very high confidence

The diagram shows the cloud architecture used to deploy the web application into Microsoft Azure. The UIA plans to deploy their system onto West US region as primary and SEA as secondary. In order to utilize the function of the system online, the database will be transferred to Azure SQL Database first which is shared between the two regions located in West US. Traffic Manager is also used to handle traffic and GitHub is used as medium to deploy into Azure.

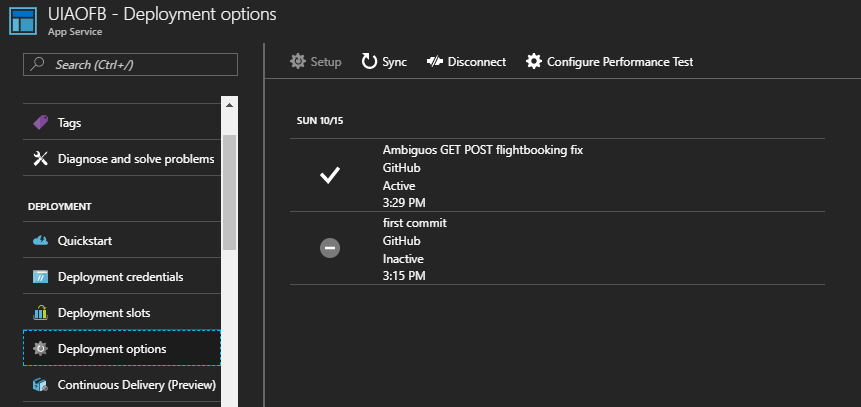
# 5.0 Implementation

## 5.1 Application Development

The system is developed using C# ASP.NET with MVC Structures and SQL Server Management Studio as database server.

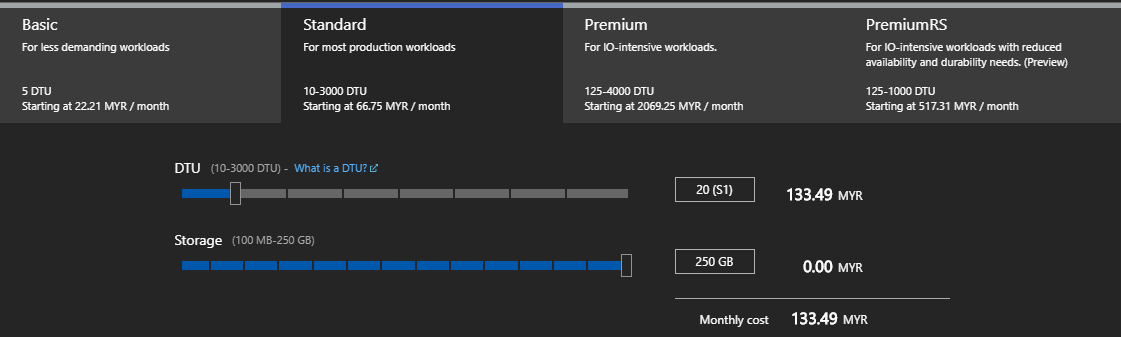
## 5.2 Azure Publishing

Two regions which is South East Asia and West US region where the application deployment is implemented with Microsoft Azure. The reason of deploying the application into two different regions is that it is to test user load on how well the traffics coming from two regions. Both regions will have the same web application deployed by using the single repository in GitHub thus when changes are made, it is only needs to be deployed once into both hosted regions. The traffic manager is also implemented to handle performance of the web application for both regions by configuring the routing method into performance option.



Database

In this application, Azure AQL Server is used as database migrated from the local database as it provides certain features which benefits the web application in terms of performance such as scaling and monitoring. The database will be used directly during the application development as it is to avoid unnecessary changes in configuration of the database.

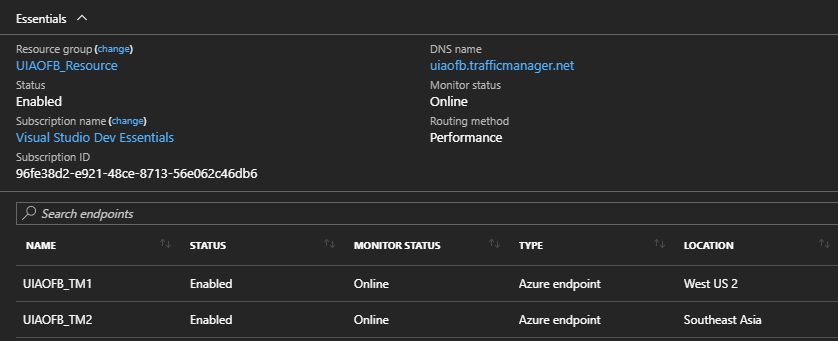


Standard S1 level Azure SQL server is used as it is sufficient to handle 2 regions. It also has 250 GB of storage for database which is sufficient to support the current needs.

## 5.3 Application Scaling

The application deployed to both regions use S1 tier App service as there are only RM 150 limit of usage per month. After the web application is presentable and ready for public use, it will then use the S3 plan or higher packages according the business needs.

All the applications use at least a Standard S1 plan or above as it has essential features such as custom domains for UIA to use. It enables for up to 10 instances to be created and auto scaled based on metrics such as server loads, and memory usage.



Traffic manager is used as it is supported by standard S1 plan. Performance is chosen for routing method to manage the best performance for user to access the application through the nearest region.

Scaling Out

The South-East Asia region will use a minimum of 1 instance due to the limited budget as more instances require higher monthly payments for scaling out. Two rules are set for an instance will be increase by 1 when the CPU percentage is above 70%, and will be reduce by 1 if the percentage is lower than 40%.



# 6.0 Test Plan

## 6.1 Unit Testing

Registration testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Title | Description | Result | |
| Expected | Actual |
| R001 | Input Validation | 1. Click register page 2. Empty required field 3. Click submit. | All fields are highlighted in red are required to fill | As expected |
| R002 | Valid Input and Sign Up | 1. Click register page 2. Type username “abc” and email [abc@hotmail.com](mailto:abc@hotmail.com). 3. Fill in other required fields with correct pattern and datatype. 4. Click submit. | Registration success and redirect user to login page. | As expected |
| R003 | Existing username | 1. Click register page 2. Type username “abc” and email [abc@hotmail.com](mailto:abc@hotmail.com). 3. Fill in other required fields with correct datatype. 4. Click submit. | Username already taken message shows up. | As expected |
| R004 | Existing email address | 1. Click register page 2. Register with username “kkk” and email [abb@hotmail.com](mailto:abb@hotmail.com). 3. Fill in other required fields with datatype. 4. Click submit. | Email is already registered message shows up. | As expected. |

Search Flight Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Title | Description | Result | |
| Expected | Actual |
| F001 | Empty search result | 1. Leave search criteria blank. 2. Clicks search. | Table shows no result. | As expected |
| F002 | Specific search result | 1. From: Thailand   To: Malaysia   1. Clicks search. | Tables shows 2 results. | As expected |
| F003 | Show flight detail | 1. From: Thailand   To: Malaysia   1. Clicks search. 2. Click on book flight of first result. | Redirects user to Flight Reservation page of flight chosen. | As expected |

Login Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Title | Description | Result | |
| Expected | Actual |
| L001 | Input Validation | 1. Click login page. 2. Leave username and password empty. 3. Clicks login. | Prompts user to enter username and password | As expected |
| L002 | Invalid login credentials | 1. Click login page. 2. Enter invalid username and password. 3. Clicks login. | Invalid username or password message shown. | As expected |
| L003 | Valid login credentials | 1. Click login page. 2. Enter correct username and password. 3. Clicks login. | Redirects user to customer home page | As expected |

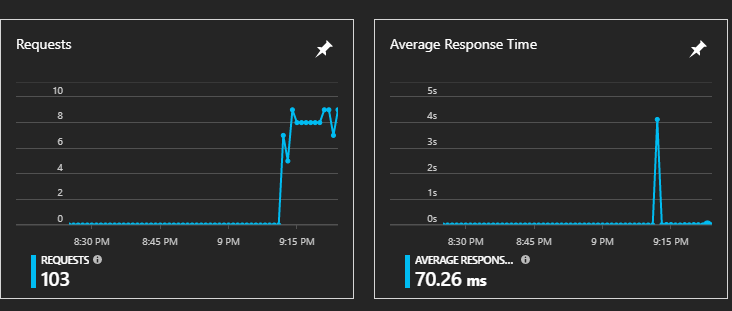
Book Flight Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Title | Description | Result | |
| Expected | Actual |
| R001 | No seat chosen | 1. Click select without choosing any seats. | Prompts user to select seats. | As expected |
| R002 | Select seats and proceed | 1. Click select after selecting seat number 1 and 2. | Passengers’ information division shows up. | As expected |
| R003 | Valid passengers’ information | 1. Fill in all passengers’ information with valid pattern after passengers’ information division shows up   **Name:** Lin  **Gender:** Male  **Nationality:** Malaysia  **DOB:** 24/6/1995  **Name:** Andy  **Gender:** Male  **Nationality:** Malaysia  **DOB:** 2/9/1992   1. Click submit. | Prompts booking success message and redirects user to View My Reservation page. | As expected |

View Reservation Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Title | Description | Result | |
| Expected | Actual |
| V001 | List entire reservations | 1. Click View My Reservations page. | All flight reservations’ records of user shown in table. | As expected |
| V002 | Search for specific reservation | 1. Enter search criteria   **From:** Thailand  **To:** Malaysia  **Departure Date:** 10/11/2017   1. Clicks search. | 1 row of result shows up. | As expected |
| V003 | View ticket and details | 1. Click on “View Detail” of first row after search. | Redirects user to reservation’s detail page of selected reservation. | As expected |

## 6.2 Performance Testing



The diagram above shows the averages response times for the usage of the application. The average time was 70.26 ms for 103 requests.

Performance test is conducted to measure the average response time for certain amounts of concurrent users. The test starts at 250 user loads to 750 user loads with increments of 250 users for 2 minutes. The data gathered for performance test is recorded in the table below.

|  |  |  |
| --- | --- | --- |
| User load | Response Time | Request |
| 250 | 0.15 | 470.88 |
| 500 | 1.39 | 360.35 |
| 750 | 1.98 | 421.88 |

# 7.0 Manage Databases

The concept of platform as a service (PaaS) is becoming a valuable tool to a market that is constantly trying to shave the time between development and deploying off. The needed resources can be purchased from a cloud service provider on a pay-as-you-go basis and access them over a secure Internet connection.

Like IaaS, PaaS includes infrastructure—servers, storage, and networking—but also middleware, development tools, business intelligence (BI) services, database management systems, and more. PaaS is designed to support the complete web application lifecycle: building, testing, deploying, managing, and updating.

PaaS allows to avoid the expense and complexity of buying and managing software licenses, the underlying application infrastructure and middleware or the development tools and other resources. The cloud service provider typically manages everything else aside from the services the developer developed.

Benefits of PaaS

1. Cut coding time

PaaS development tools can cut the time it takes to code new apps with pre-coded application components built into the platform, such as workflow, directory services, security features, search, and so on.

1. Add development capabilities without adding staff

Platform as a Service components can give the development team new capabilities without the needing to add staff having the required skills.

1. Develop for multiple platforms—including mobile—more easily

Some service providers give you development options for multiple platforms, such as computers, mobile devices, and browsers making cross-platform apps quicker and easier to develop.

1. Use sophisticated tools affordably

A pay-as-you-go model makes it possible for individuals or organizations to use sophisticated development software and business intelligence and analytics tools that they could not afford to purchase outright.

1. Support geographically distributed development teams

Because the development environment is accessed over the Internet, development teams can work together on projects even when team members are in remote locations.

1. Efficiently manage the application lifecycle

PaaS provides all of the capabilities that are needed to support the complete web application lifecycle: building, testing, deploying, managing, and updating within the same integrated environment.

# 8.0 Conclusion

In conclusion, the development of a cloud based application for Ukraine International Airlines has proven to be a challenging albeit informative one. The process of not just building but also deploying an application fit for the cloud and one that utilizes all the cloud has to offer has provided new, relevant skills which will be invaluable to the modern informational technology industry. Concepts such as cloud design patterns and cloud resource provisioning as well as the considerations needed to design a cloud architecture have become much clearer. A deeper understanding of the underlying technology of cloud computing through Azure has also been obtained. Hence, this project has been a very productive one and I look forward to more like it in the future.

# 9.0 References

Amazon Web Services, 2013. *Amazon EC2 Service Level Agreement.* [Online]   
Available at: https://aws.amazon.com/ec2/sla/  
[Accessed 4 November 2017].

Amazon Web Services, 2017. *Amazon RDS Product Details.* [Online]   
Available at: https://aws.amazon.com/rds/details/  
[Accessed 4 November 2017].

AnyChart.com, 2017. *AnyChart.* [Online]   
Available at: https://www.anychart.com/  
[Accessed 4 November 2017].

Apache Friends, 2017. *XAMPP Apache + MariaDB + PHP + Perl.* [Online]   
Available at: https://www.apachefriends.org/  
[Accessed 5 November 2017].

Cavale, A., Boucher, R. J. & Kemnetz, J., 2017. *Scale instance count manually or automatically.* [Online]   
Available at: https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/insights-how-to-scale  
[Accessed 5 November 2017].

Engine Yard, 2017. *Top 10 Advantages of Platform as a Service.* [Online]   
Available at: http://www.engineyard.com/whitepapers/top-10-advantages-of-platform-as-a-service  
[Accessed 5 November 2017].

generatedata.com, 2017. *generatedata.com.* [Online]   
Available at: https://www.generatedata.com/  
[Accessed 6 November 2017].

GitHub Inc., 2017. *GitHub.* [Online]   
Available at: https://github.com/  
[Accessed 6 November 2017].

Google, 2015. *Google Compute Engine uses Live Migration technology to service infrastructure without application downtime.* [Online]   
Available at: https://cloudplatform.googleblog.com/2015/03/Google-Compute-Engine-uses-Live-Migration-technology-to-service-infrastructure-without-application-downtime.html  
[Accessed 10 November 2017].

Google, 2017. *Google Infrastructure Security Design Overview.* [Online]   
Available at: https://cloud.google.com/security/security-design/  
[Accessed 11 November 2017].

Hao, L., H, J. & Hubbard, J., 2017. *Explaining Compute Units in Azure Database for MySQL.* [Online]   
Available at: https://docs.microsoft.com/en-us/azure/mysql/concepts-compute-unit-and-storage  
[Accessed 11 November 2017].

Lin, C., 2016. *Scale up an app in Azure.* [Online]   
Available at: https://docs.microsoft.com/en-us/azure/app-service/web-sites-scale  
[Accessed 12 Novemberber 2017].

Luijbregts, B., 2017. *How to Autoscale Azure App Services & Cloud Services.* [Online]   
Available at: https://stackify.com/autoscale-azure-app-services-cloud-services/  
[Accessed 12 November 2017].

Microsoft, 2017. *App Service plans.* [Online]   
Available at: https://azure.microsoft.com/en-us/pricing/details/app-service/plans/  
[Accessed 13 November 2017].

Microsoft, 2017. *Azure Database for MySQL.* [Online]   
Available at: https://azure.microsoft.com/en-us/services/mysql/  
[Accessed 13 November 2017].

Microsoft, 2017. *Microsoft Azure Documentation.* [Online]   
Available at: https://docs.microsoft.com/en-us/azure/  
[Accessed 17 November 2017].

Narumoto, M., Bennage, C. & Wasson, M., 2017. *Health Endpoint Monitoring Pattern.* [Online]   
Available at: https://docs.microsoft.com/en-us/azure/architecture/patterns/health-endpoint-monitoring  
[Accessed 19 November 2017].