

|  |  |  |
| --- | --- | --- |
| **Module Code** | **:** | CT071-3-5-3-DDAC |
| **Intake Code** | **:** | UC3F1702SE |
| **Student Name** | **:** | Bryton Hong Jun Mun |
| **TP No** | **:** | TP034139 |
| **Lecturer Name** | **:** | Dr. Kalai Anand Ratnam |
| **Hand out Date** | **:** | 12 JULY 2017 |
| **Hand in Date** | **:** | 20 NOVEMBER 2017 |
| **Assignment Title** | **:** | Online Flight Booking System |

# Acknowledgement

I would like to take this opportunity to express my gratitude to Dr. Kalai Anand Ratnam who have taught and provided me with resources such as reading materials, and lab activities which lead to the completion of this assignment. Through his teachings in designing and developing cloud applications, I was introduced to Microsoft Azure which benefited me greatly because as time passes, the usage and importance of cloud computing in the information technology industry is ever growing. In addition to the Dr. Kalai’s teachings, implementation of certain Microsoft Azure features which was essential for the development of this assignment was made possible through his detailed explanations. Besides my lecturer, I would also like to express my gratitude to my friends for providing me with constructive feedback.

**Table of Contents**

[Acknowledgement 1](#_Toc498941016)

[1) Introduction 4](#_Toc498941017)

[1.1) Background 4](#_Toc498941018)

[1.2) Aims and Objectives 4](#_Toc498941019)

[1.2.1) Aims 4](#_Toc498941020)

[1.2.2) Objectives 4](#_Toc498941021)

[1.3) Requirement Specifications 4](#_Toc498941022)

[1.4) Deliverables 4](#_Toc498941023)

[2.0) Project Plan 5](#_Toc498941024)

[3.0) Design 6](#_Toc498941025)

[3.1) Architectural Diagram 6](#_Toc498941026)

[3.2) Design Consideration 7](#_Toc498941027)

[3.3) Modelling 7](#_Toc498941028)

[3.3.1) Use Case Diagram 7](#_Toc498941029)

[3.3.2) Sequence Diagram 8](#_Toc498941030)

[3.3.3) Data Modelling 10](#_Toc498941031)

[3.3.4) Sitemap 10](#_Toc498941032)

[3.4) Web Application User Interface 11](#_Toc498941033)

[4.0) Implementation 14](#_Toc498941034)

[4.1) Application Development 14](#_Toc498941035)

[4.2) Azure Deployment 16](#_Toc498941036)

[4.2.1) Creating SQL Server 16](#_Toc498941037)

[4.2.2) Creation of Web Service 18](#_Toc498941038)

[4.2.3) Publishing the Application to Azure 19](#_Toc498941039)

[4.2.4) Creation of Traffic Manager Profile 21](#_Toc498941040)

[4.3) Application Scaling 22](#_Toc498941041)

[4.3.1) SQL Database Pricing Tier 22](#_Toc498941042)

[4.3.2) Web App Service Pricing Tier 22](#_Toc498941043)

[4.4) Investigate and Analyse Application 24](#_Toc498941044)

[4.5) Implementation and Discussion of Managed Database 26](#_Toc498941045)

[5.0) Test Plan and Testing Discussion 28](#_Toc498941046)

[5.1) Unit Testing 28](#_Toc498941047)

[5.2) Performance Testing 30](#_Toc498941048)

[6.0) Conclusion 32](#_Toc498941049)

[7.0) References 33](#_Toc498941050)

# 1) Introduction

## 1.1) Background

An Online Flight Booking System has been developed for Ukraine International Airlines (UIA). The Online Flight Booking System provides several features for users such as user registration, flight searching, flight booking, and flight booked viewing. Based on those features, the system will be required to utilize a database to store various information regarding the reservations of flight, flight details, airport details, and user details to provide commercial operations to UIA. Rules and regulations were implemented for features to business logic of the system. In addition to that, the system was published through Microsoft Azure and has two SQL Databases, the primary database which functions as the initial database utilized by the system and the secondary database functions as the backup database in case the primary database crashes.

## 1.2) Aims and Objectives

### 1.2.1) Aims

To provide Ukraine International Airlines an online platform which will allow flight booking.

### 1.2.2) Objectives

1. To design the user interface of the online flight booking system
2. To develop the online flight booking system via Visual Studio
3. To implement database features from Microsoft Azure to the online booking system
4. To publish the online booking system through Microsoft Azure

## 1.3) Requirement Specifications

1. Allow user to register and login into the system
2. Allow users to search for existing flight details
3. Allow users to book their desired flight
4. Allow users to view their booking history

## 1.4) Deliverables

1. Design & Develop a web application hosted on Microsoft Azure as a web application.
2. To utilize SQL Database to store data recorded from the system
3. To analyse web application performance with monitoring tools.
4. To scale the solution to meet the needs of demands during peak season

# 2.0) Project Plan

The Gantt Chart above represents the schedule of the design and development of the online booking system. It provides details of the timeline for each phase and stages throughout the progress of the assignment such as the Introduction phase, Design phase, Implementation phase, and Testing phase, Project Documentation and the Conclusion of the assignment. In addition to that, each phase has its own sub phases of the duration required to complete it.A screenshot of a cell phone

Description generated with high confidence

# 3.0) Design

## A close up of a map Description generated with high confidence3.1) Architectural Diagram

Diagram 1: Cloud Architecture Diagram

Diagram 1 illustrates the cloud architecture implemented within the Online Flight Booking System which is published through Microsoft Azure platform. The developed flight booking system has been deployed to two regions, which South East Asia and Central US. Due to the implementation of the Traffic Manager profile, the User will only be able to access the Web App which location is the nearest to the user’s location. For example, should the user be located in or near South East Asia, the user will only be able to access the Web App within the South-East Asia region whereas if the user’s location is either within or near West US. Two SQL server were implemented, primary and secondary. The Primary SQL Server is utilized by the Web App from both the South-East Asia and West US region, is located within South East Asia and functions as the main database for storing information whereas the Secondary SQL Server which is located in West US 2 functions as a backup in the event that the primary server crashes.

## 3.2) Design Consideration

During the design considerations, several assumptions were determined. It was determined that UIA has desired to spread its business to customers beyond Ukraine and there will require the deployment of web app into different regions to cover a wide range of customers from different countries. By utilizing Microsoft Azure, the development of the web application has a fixed budget of RM 150 and the cost of required features will have to be assessed. The Required features will be based on the features available on Azure which is necessary for the Flight Booking system to provide its services. Such required features that are available on Azure are the SQL Servers and Database, Traffic Manager, and Web App services which has a variety of service plans.

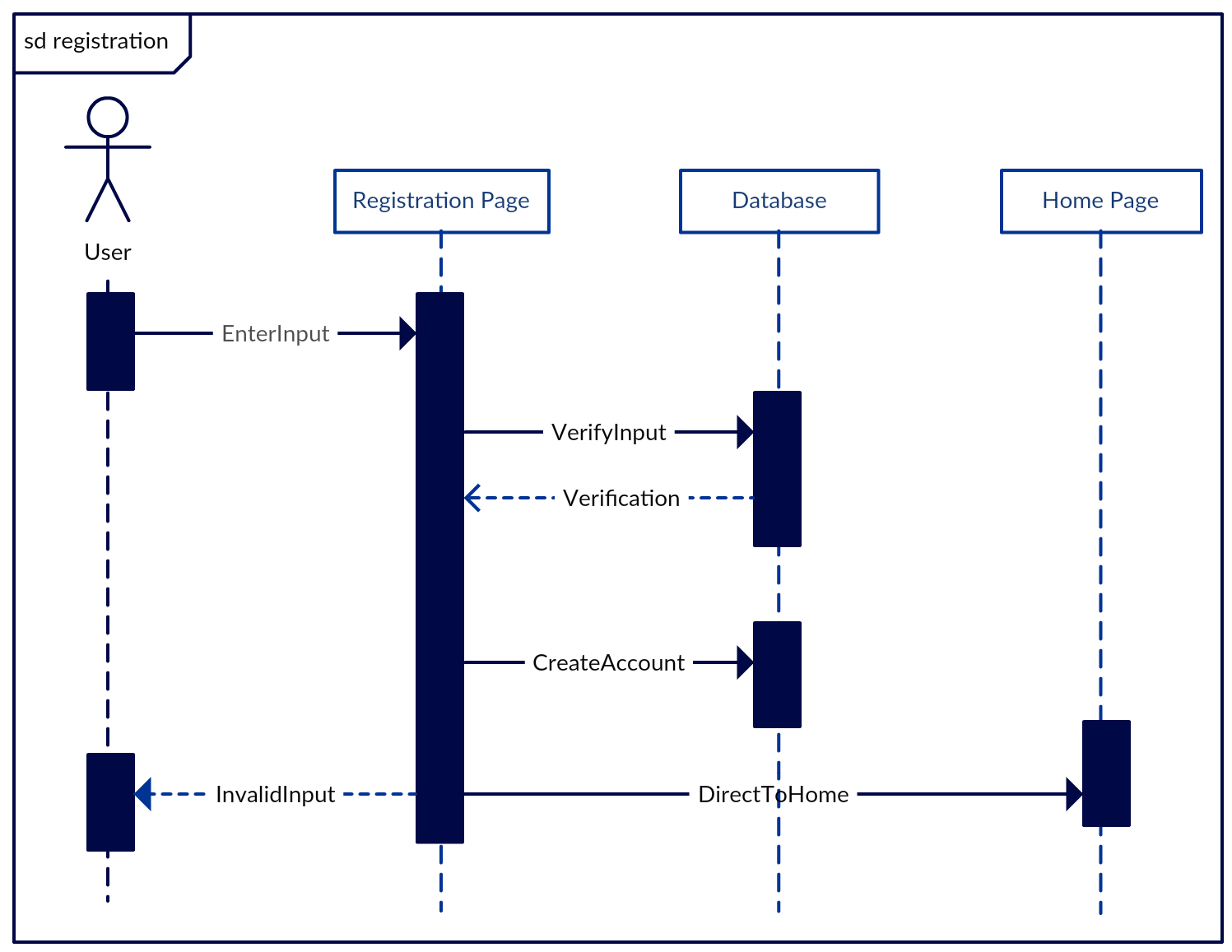
## 3.3) Modelling

### A close up of a map Description generated with very high confidence3.3.1) Use Case Diagram

Diagram 2: Use Case

As shown in Diagram 2: Use Case, the unregistered user will only be able to register into the system and will not be able to login, search for flight, book flight, or view the booking details as their details are not recorded within the system. The registered user will be able to login to the system, search for available flights, book the available flights, and view their booked flights.

### 3.3.2) Sequence Diagram

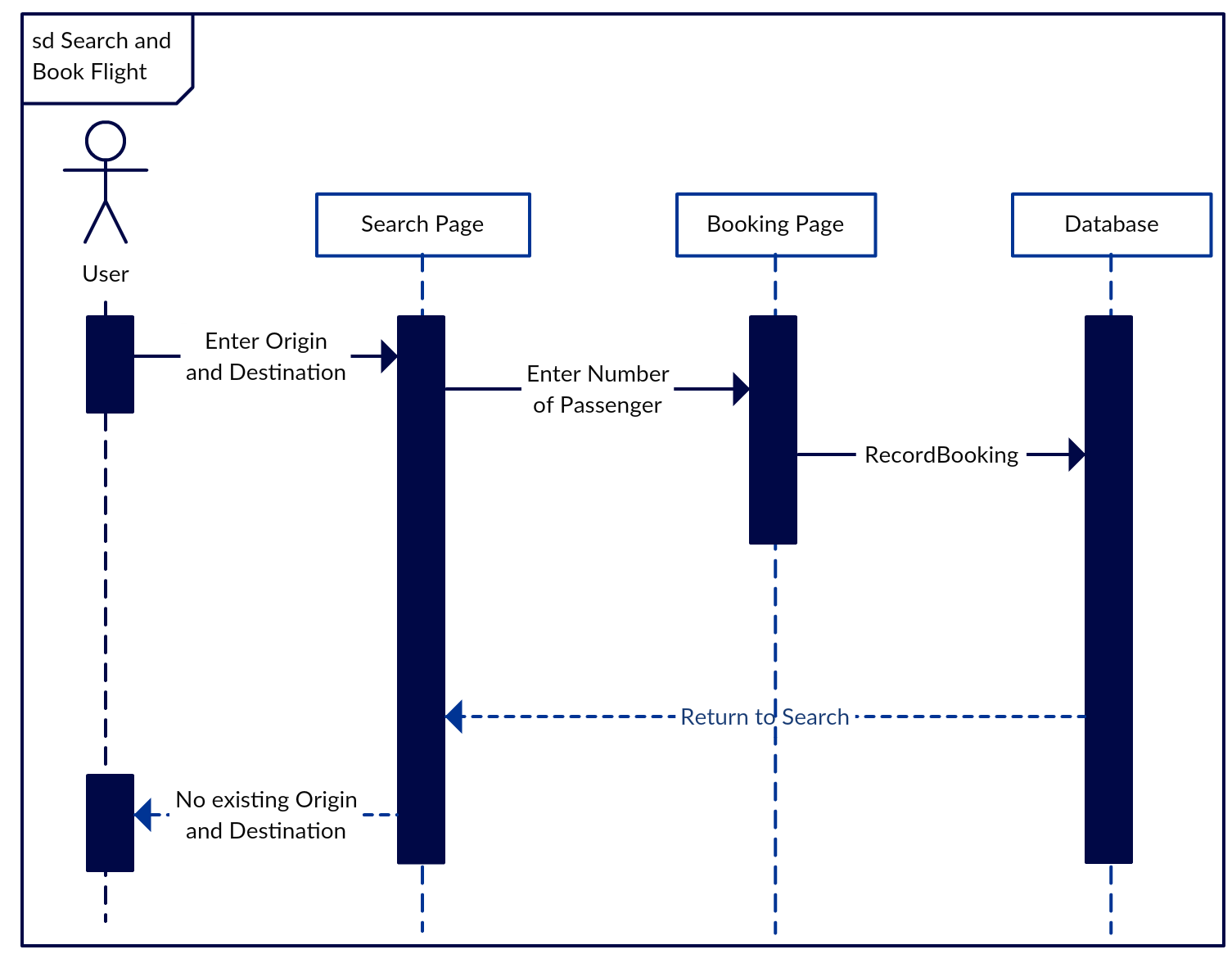
The Sequence diagrams below illustrates the logical flow within the online flight booking system.

Sequence Diagram: Registration

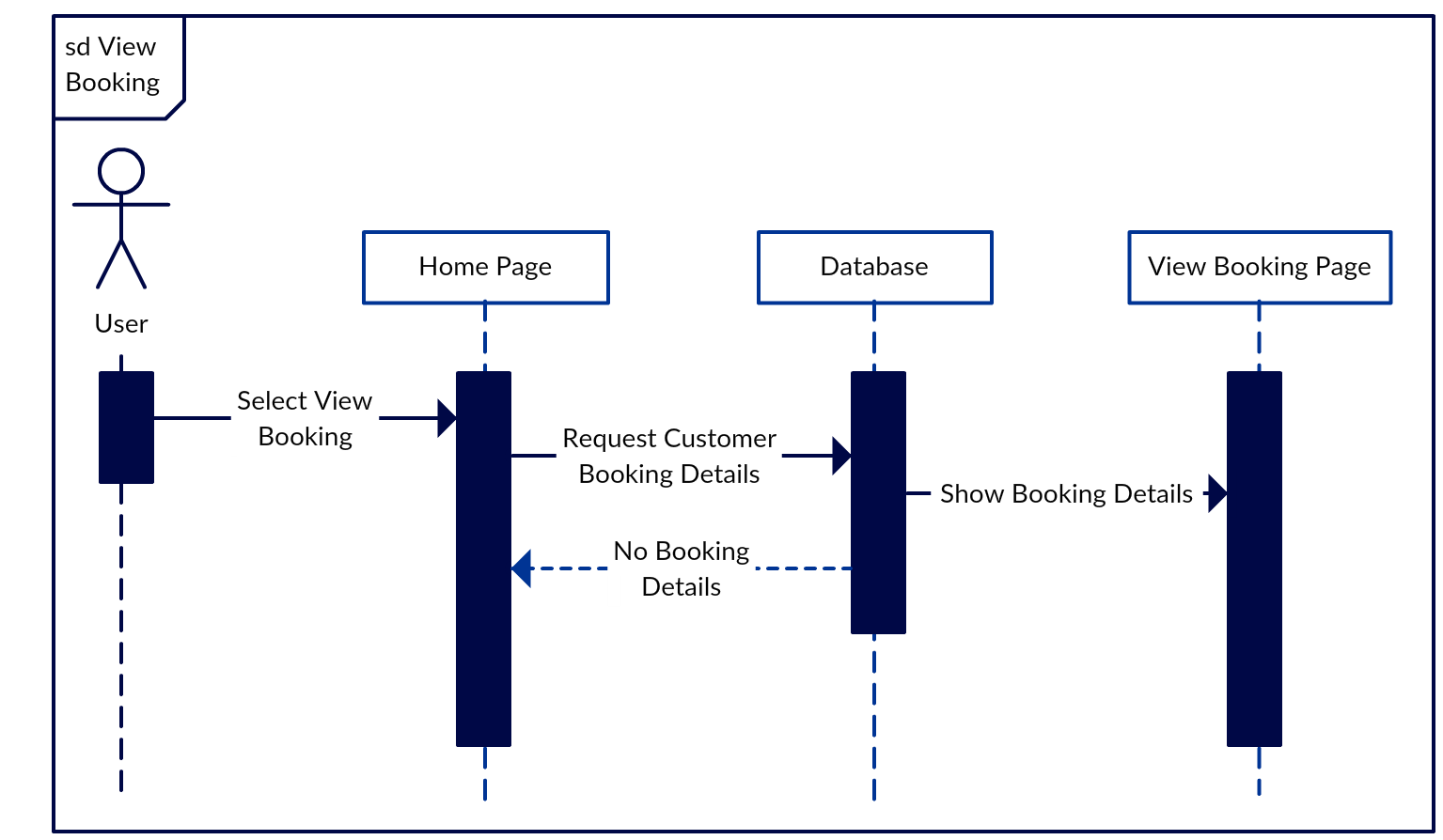
A screenshot of a cell phone

Description generated with very high confidence

Sequence Diagram: Login



Sequence Diagram: Search Flight



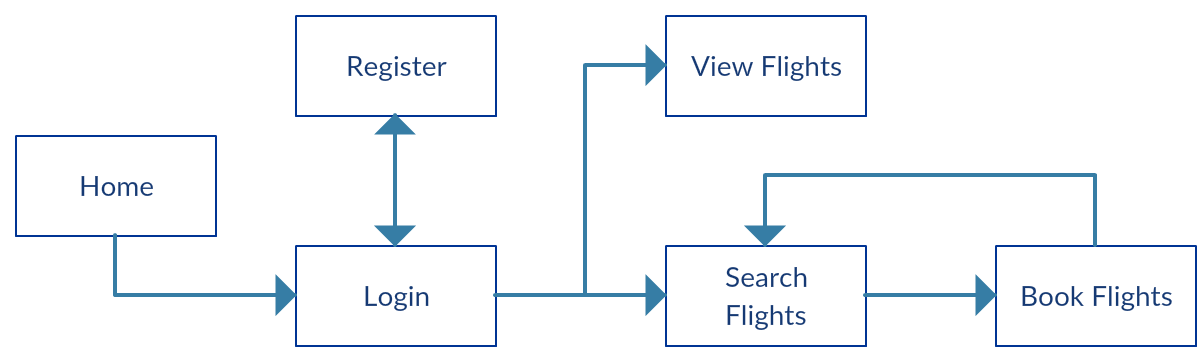
Sequence Diagram: Search Flight

### 3.3.3) Data Modelling

Entity Relationship Diagram

The Entity Relationship Diagram above illustrates the relationship between the tables, Airports, Flights, Flight\_Booking, and AspNetUsers. The “arrive\_airport\_id” and “origin\_airport\_id” are the foreign keys within Flights and are retrieved from the “airport\_id” in Airports table. The “selected\_flight\_id” and “ApplicationUser\_Id” are the foreign keys within Flight\_Booking in which “selected\_flight\_id” is retrieved from “flight\_id” in Flights table whereas the “ApplicationUser\_Id” is retrieved from id within the “AspNetUsers” table. The Airports table has a Many-to-Many relationship with Flights table, whereas the Flights table has Many-to-Many relationship with Flight\_Booking and Flight\_Booking has a Many-to-One relationship with AspNetUsers.

### 3.3.4) Sitemap



The Sitemap above illustrates the list of webpages and their flow between one another. The Users will first be required to login. If the user is an unregistered user, he or she will be required to register. Once that, the registered will be able to search, book, and view flights.

## A screenshot of a cell phone Description generated with very high confidence3.4) Web Application User Interface

UI: Login Page

A screenshot of a social media post

Description generated with very high confidence

UI: Registration Page

A screenshot of a cell phone

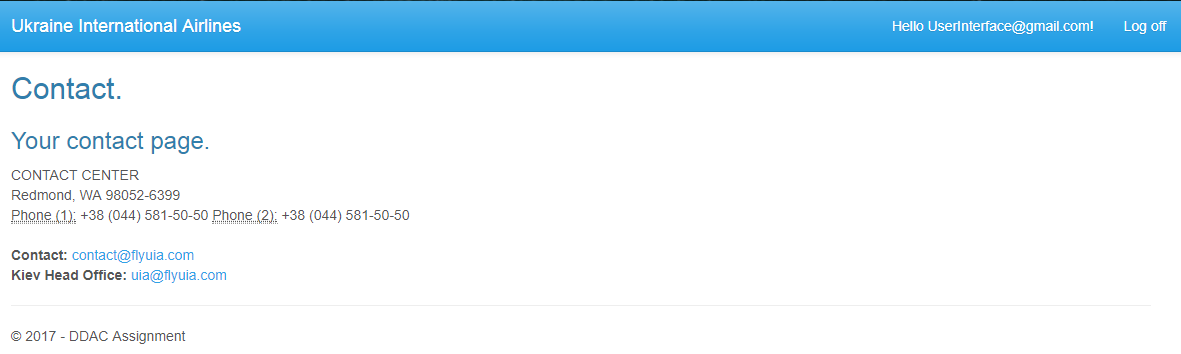
Description generated with very high confidence

UI: Home Page

A screenshot of a social media post

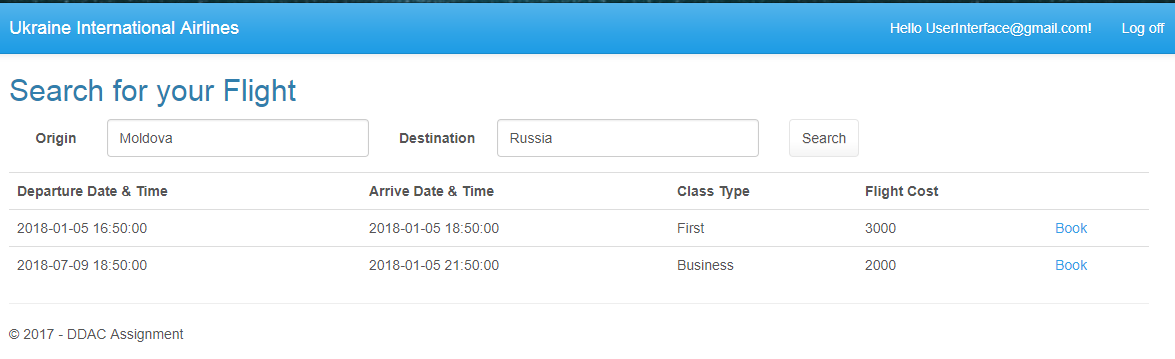
Description generated with very high confidence

UI: About Page



UI: Contact Page

A screenshot of a social media post

Description generated with very high confidence

UI: Book Flight Page

UI: Search Flight Page

A screenshot of a social media post

Description generated with very high confidence

UI: View Flight Page

4.0) Implementation

## 4.1) Application Development

A screenshot of a cell phone

Description generated with very high confidenceThe programming language utilized to develop the Online Flight Booking System was C# and HTML. In addition to that the development tool used to develop the web application was Visual Studio 2017 and the database tool used was Microsoft SQL Server Management Studio. The development of the web application is based on the web application framework provided my Microsoft, which is ASP.NET MVC. This framework implements the model-view-controller (MVC) pattern within the application. When creating the project, the MVC template was selected and the Authentication of the web application was changed to Individual Authentication as shown in Image 1.

Image 1

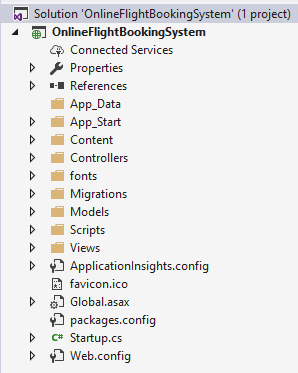
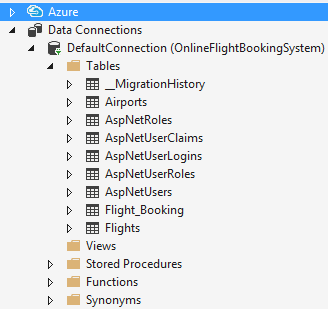


Image 2 shows the folders within the created project. Furthermore, since the web application was created with MVC selected, the Model folder will contain all the code files for managing the web application data, the Controller folder will contain code files to send commands to the files within the Model folder, and the View folder which displays results of the user’s action. The code files within the Model and Controller folder will be written in C# and the code files within the View Model is written in HTML.

Image 2

The web application data was stored within Microsoft SQL Server Management Studio initially, before it was connected to SQL Database within Microsoft Azure. During the phase of development, the web application utilized a created database called, UIA\_FlightDatabase, to store data submitted by users. Image 3 shows the data connections from Visual Studio and Image 4 shows the data connections within SQL Server Management Studio. The Database A screenshot of a cell phone

Description generated with very high confidencewithin the SQL Server Management Studio will allow the developer to view stored data.

Image 4

Image 3

The completed web application is available on GitHub simply by using the link provided: **https://github.com/BrytonHong/DDAC-Assignment**

## 4.2) Azure Deployment

### 4.2.1) Creating SQL Server

A screenshot of a cell phone

Description generated with very high confidenceBefore the web application is published to the Microsoft Azure, a SQL Database will have to be created within the Azure as shown in Image 5.

Image 5

**A close up of a map

Description generated with high confidence**During the creation of the SQL Database, a Resource Group will be created along with it. Both the SQL Database and SQL Server will have been stored within the Resource Group and will be located within South East Asia. The role of the first created SQL Database is the primary the database. Once the SQL Database has been created, a Geo-Replication will be created so as to provide a secondary database which will function as a backup in the event that the primary database crashes. Given that the primary database is located within South East Asia, the secondary database will be located within West US 2 as shown in Image 6.

Image 6

Image 6

After the creation of the SQL Database, the database will automatically generate a Connection String to be copy and pasted into the Web Config file of the web application project as shown in Image 7. To Connection can be retrieved from the “Connection Strings” which is located under Settings of the select database which in this case is the primary database. The connection string is as shown in Image 8. The User ID username and password will have to be changed according to the username and password created during the creation of the SQL Database.

**A screenshot of a cell phone

Description generated with high confidence**A screenshot of a social media post

Description generated with very high confidence

Image 8

Image 7

### 4.2.2) Creation of Web Service

A screenshot of a cell phone

Description generated with very high confidenceAfter the process of creating the SQL Database, a Web App as shown in Image 9 from Azure must be created to allow the Online Flight Booking System to be published through it.

Image 9

During the creation the Web App service in Azure, an App Name must be given and utilizes an existing Resource Group which was created during the creation of the SQL Server. During this phase, the Application Insights as shown in Image 10 will be turned on in order to allow the web application to be monitored in terms of the performance. After all necessary information has been entered, the web app service will be available to allow the flight booking system to be published through Visual Studio

A screenshot of a cell phone

Description generated with very high confidence

Image 10

### 4.2.3) Publishing the Application to Azure

A screenshot of a cell phone

Description generated with very high confidenceOnce the Web App service has been created on Azure, the Online Flight Booking System is published from Visual Studio by selecting “Publish” on the web application project under the Solution Explorer as shown in Image 11.

Image 11

A screenshot of a cell phone

Description generated with very high confidenceAfter that, an option will be prompted in which the options “Create New” or “Select Existing” as shown in Image 12. Since an Azure Web App service has already been created, the “Select Existing” will be chosen.

Image 12

A screenshot of a cell phone

Description generated with very high confidenceAfter “Select Existing” has been selected, a window will be prompted to allow selection of an existing app service from a Resource Group as shown in Image 13. Once the desired App Service has been selected it will appear as shown in Image 14 as it will be ready to be published.

Image 13

A screenshot of a cell phone

Description generated with very high confidence

Image 14

### 4.2.4) Creation of Traffic Manager Profile

A screenshot of a cell phone

Description generated with very high confidenceAfter the web application has been published, a Traffic Manager profile as shown in Image 15, was implemented to allow the User to access the Web App which location is the nearest to the user’s location. For example, should the user be located in or near South East Asia, the user will be able to access the Web App within the South-East Asia region and should another user who is located within or near West US, he or she will be able to access the West US version of the web application.

Image 15

A screenshot of a social media post

Description generated with very high confidenceOnce the Traffic Manager profile has been created, two endpoints will be created within the Traffic Manager. One of the Endpoint would be assigned to the South-East Asia region and the other endpoint will be assigned to the West US region as shown in Image 16.

Image 16

## 4.3) Application Scaling

### 4.3.1) SQL Database Pricing Tier

A screenshot of a cell phone

Description generated with very high confidenceThe deployment of the Online Flight Booking System will be using the Basic pricing for the creation of both the SQL Database as shown in Image 17. The Basic pricing of the SQL Database has a monthly cost of RM 22.21, a storage of 2 Gigabyte, and 5 Data Transaction Units. Considering these factors and given the fact that the size of data storage for the web application is not large and the budget allocated to the development of the web application is fixed to only RM 150 a month it is practical to select the Basic Pricing Tier. Should the utility of the web application database be increased, the pricing plan can be easily upgraded to suit the business needs.

Image 17

### 4.3.2) Web App Service Pricing Tier

Since the deployment of the Online Flight Booking System required the Web App service, the creation of the Web App service required the selection of a pricing tier. Given the factors as stated within the **4.3.1) SQL Database Pricing Tier**, the selection of the App Service Plan will be S1 Standard which will cost an estimated RM 331.08 per month and is the cheapest Plan amongst the others as shown in Image 18. Besides the reason of the low cost, the plan also consist the essential features such as 1 Core, 1.75 GB RAM, and allows the App Service to utilize Traffic Manager Geo availability which is appropriate for the web application. Given that the features of the plan provides every required feature for the web application to provide its business needs and user satisfaction and that the budget limit for the deployment of the web application is RM 150, it has been determined it would be practical to select the S1 Standard.

A screenshot of a cell phone

Description generated with very high confidence

Image 18

## 4.4) Investigate and Analyse Application

Image 20

Image 19

A screenshot of a cell phone

Description generated with very high confidenceA screenshot of a cell phone

Description generated with very high confidenceIn Image 19 it shows the Data In graph which illustrates the quantity of incoming bandwidth used up by the web application in MiBs (Microsoft Azure, 2016). Based on the results in Image 19, it displays a high rise between 4am and 4:15am between 30kB and 40kB and has recorded 178.21 kB od data and a drop after 4:15am which means the number of users has decreased.. Within Image 20, it shows the Data Out graph, which illustrates the quantity of outgoing bandwidth used up by the app in MiBs (Microsoft Azure, 2016). Based on the results in Image 20, it displays a high rise between 4am and 4:15am between 30kB and 40kB and has recorded 198.28 kB and a drop after 4:15am which means the number of users has decreased.

Image 21

Image 22

In Image 21 it shows the Request graph which illustrates the total number of requests regardless of their resulting HTTP status code. (Microsoft Azure, 2016). Based on the results in Image 19, it displays a high rise between 4am and 4:15am between 20 and 25 and has recorded 107 requests and a drop after 4:15am which means the number of users has decreased. Within Image 22, it shows the Average Response Time graph, which illustrates the average time taken for the app to serve requests in ms. (Microsoft Azure, 2016). Based on the results in Image 20, it displays a high rise between 4am and 4:15am between 6s and 8s and has recorded 191.77 ms and a drop after 4:15am which means the number of users has decreased. In Image 23, it illustrates the the HTTP Server errors that occurred during deployment of the web application. Based on the results within Image 23, it has not recorded any errors during deployment.

A screenshot of a cell phone

Description generated with very high confidence

Image 23

## 4.5) Implementation and Discussion of Managed Database

Cloud computing is a method for carrying data knowledge (IT) services in that resources are retrieved from the Internet across web-based instruments and requests, as challenged to a manage connection to a server (Investopedia, n.d.). Rather than keeping files on a proprietary hard drive or innate storage mechanism, cloud-based storage makes it probable to save them to a remote database. As long as an electronic mechanism possesses admission to the web, it possesses admission to the data and the multimedia plans to run it. It is considered “cloud” computing because the data being accessed is discovered in "the cloud" and does not need a user to be in a specific locale to gain admission to it (Investopedia, n.d.). This kind of arrangement permits operatives to work remotely. Firms bestowing cloud services enable users to store files and requests on remote servers, and next admission all the data via the internet. This is proven in the growth graph in Image 24.

A screenshot of a cell phone

Description generated with very high confidence

Image 24: Cloud Computing Market Growth (Forbes.com)

A screenshot of a cell phone

Description generated with very high confidence

Image 25: Range of PaaS (Microsoft Azure.com)

**Platform as a Service (PaaS)** is a complete progress and placement nature in the cloud, alongside resources that enable users to hold everything from easy cloud-based apps to urbane, cloud-enabled enterprise applications. Clients can circumvent the demand to buy multimedia or servers, and instead buy these resources in an outsourced, on-demand service. The utilization of PaaS provides many advantages such as **Cut Coding Time.** This is because progress instruments can cut the period it seizes to program new apps alongside pre-coded request constituents crafted into the period, such as workflow, directory services, protection features, find, and so on (Microsoft Azure.com). Besides that, advantage is **enhanced development capabilities without adding staff**. This is because PaaS constituents can give your progress team new skills lacking your demanding to add workers possessing the needed skills. Furthermore, will be the advantage of **Efficiently manage the application lifecycle.** This is because PaaS provides all of the skills that is demanded to prop the finished web request lifecycle such as constructing, assessing, employing, grasping, and notifying inside the alike consolidated environment (Microsoft Azure.com).

# 5.0) Test Plan and Testing Discussion

## 5.1) Unit Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Login Testing** | | | |
| **No.** | **Description** | **Expected Results** | **Actual Results** |
| 1 | Registered Email and Password is entered | User is logged into the System | Home page is displayed for the registered user |
| 2 | Unregistered Email and Password is entered | Message prompts which informs the users that he or she is not registered | Home page is not displayed for the user |
| **Registration Testing** | | | |
| **No.** | **Description** | **Expected Results** | **Actual Results** |
| 1 | Unregistered user enters the required data for registration and submits | User is then registered and prompted into the system Home page | User is successfully registered and is prompted into the system Home page |
| 2 | Unregistered user enters the incorrect data for registration and submits | Message prompts informing the user of the incorrect data and prevents user registration | User is not registered and message prompts to inform the user of the incorrect inputs |
| **Search Flight Testing** | | | |
| **No.** | **Description** | **Expected Results** | **Actual Results** |
| 1 | User searches for flight by entering existing flight origin and flight destination | Flight details is displayed | Flight details is displayed |
| 2 | User searches for flight by entering non-existing flight origin and flight destination | Flight details is not displayed | Flight details is not displayed |
| **Book Flight Testing** | | | |
| **No.** | **Description** | **Expected Results** | **Actual Results** |
| 1 | User selects desired flight to be booked | User is prompted into the Booking page with the flight details displayed | User is redirected into the Booking page and is able to view selected flight details |
| 2 | User enters the number of passengers for the selected flight to be booked | System records the number of passengers and multiplies it with the cost of the flight | System registers the booked details which includes number of passengers and total cost of flight |
| **View Booked Flights Testing** | | | |
| **No.** | **Description** | **Expected Results** | **Actual Results** |
| 1 | User selects the view flights button | User is redirected into View Booking Page and system displays booked flight details | User is prompted into the View Booking page and is able to view the displayed booked flights |

## A screenshot of a cell phone Description generated with very high confidence5.2) Performance Testing

Image 26

A close up of a logo

Description generated with high confidenceLocated within the created Web App Service which is UIAFlightBookingWebSEA, the Performance Test is located under the Development Tools. As shown in Image 26, the details will have to be entered in order to create the performance test. The Configure Test utilized is the manual test. Besides that, load form is generated based on the selected web app location is which is South East Asia, the input for user load which means virtual load is 250, and the duration of the performance test is 5 minutes.

Image 27

Once the Performance Test has been concluded, the results of the successful and failed request will be identified as shown in Image 27. Besides that, the performance of the web application has determined that with 250 users, the average request time per sec is 1.04 and the request per sec is 229.81 as shown in Image 28.

A screenshot of a cell phone

Description generated with high confidence

Image 28

The summary of the performance test results are shown in the table below.

|  |  |
| --- | --- |
| **Name** | **PerfTest01** |
| **User Load** | 250 |
| **Request (Successful)** | 689400 (100%) |
| **Request (Failed)** | 0 (0%) |
| **Average Request Time (Sec)** | 1.04 |
| **Request (Sec)** | 229.81 |

# 6.0) Conclusion

In conclusion, the Online Flight Booking System was developed with several features such as login, registration, search existing flights based on origin and destination, book flight by selecting the desired flight and entering the number of passengers, and viewing the booked flights. The development of the system utilized programming languages such as C# for functions and HTML for user interface. The development tool utilized was Visual Studio 2017. Furthermore, Microsoft SQL Server Management Studio was utilized as the initial SQL database platform before the system was deployed into Azure. After the system was deployed into Azure, it utilized the SQL Service provided by Azure. The process of designing, developing, deploying the web application has greatly benefited me. This is because I have learned much on the advantages of utilizing cloud services such as being able to avoid large cost of running SQL servers physically as cloud computing is a virtual data storage and provides an all around the clock service.

In addition to benefits of cloud services, there are many advantages to Microsoft Azure besides just providing cloud services for deploying the web application. An example of this is the Traffic Manager Feature which will allow the web application to appear differently to users accessing it based on the users’ regional location, which can be extremely useful for businesses which seeks to provide satisfactory services to customers based on their background. Besides that, Microsoft Azure also provides features such as Performance testing on the web application, and monitoring tools to manage and keep track of the web application’s request, process of data, and occurrence of error as it is running.

# 7.0) References

1. Investopedia. 2017. Cloud Computing. [ONLINE] Available at: https://www.investopedia.com/terms/c/cloud-computing.asp. [Accessed 19 November 2017].
2. Microsoft Azure. 2017. What is PaaS? Platform as a Service | Microsoft Azure. [ONLINE] Available at: https://azure.microsoft.com/en-us/overview/what-is-paas/. [Accessed 19 November 2017].
3. Louis Columbus. 2017. Roundup Of Cloud Computing Forecasts, 2017. [ONLINE] Available at: https://www.forbes.com/sites/louiscolumbus/2017/04/29/roundup-of-cloud-computing-forecasts-2017/#4dbc1d4731e8. [Accessed 7 November 2017].