****

**INDIVIDUAL ASSIGNMENT**

**TECHNOLOGY PARK MALAYSIA**

**DESIGNING AND DEVELOPING APPLICATION ON THE CLOUD**

**CT071-3-3-DDAC**

**UC3F1701SE**

**DATE ASSIGNED: 3rd July 2017**

**DATE COMPLETED: 9th Oct 2017**

**WEIGHTAGE: 100%**

**INSTRUCTIONS TO CANDIDATES:**

1. **Submit your assignment at the administrative counter.**
2. **Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing).**
3. **Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.**
4. **Cases of plagiarism will be penalized.**
5. **The assignment should be bound in an appropriate style (comb bound or stapled).**
6. **The assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.**

**7. You must obtain 50% overall to pass this module.**

Table of Contents

[Introduction 3](#_Toc495314913)

[Project Plan 4](#_Toc495314914)

[Design 5](#_Toc495314915)

[Cloud Design Patterns 5](#_Toc495314916)

[Architectural Design 6](#_Toc495314917)

[Design Considerations 7](#_Toc495314918)

[Modelling 7](#_Toc495314919)

[Use Case Diagram 7](#_Toc495314920)

[Context Diagram 8](#_Toc495314921)

[DFD level 1 8](#_Toc495314922)

[Page Flow Chart 9](#_Toc495314923)

[Entity Relationship Diagram 9](#_Toc495314924)

[Implementation 10](#_Toc495314925)

[Test Plan 11](#_Toc495314926)

[Unit testing 11](#_Toc495314927)

[Pages Navigation 11](#_Toc495314928)

[Login page 11](#_Toc495314929)

[Register page 12](#_Toc495314930)

[Booking page 12](#_Toc495314931)

[View booking page 13](#_Toc495314932)

[Logout 13](#_Toc495314933)

[Performance Test 14](#_Toc495314934)

[Conclusion 17](#_Toc495314935)

[References 18](#_Toc495314936)

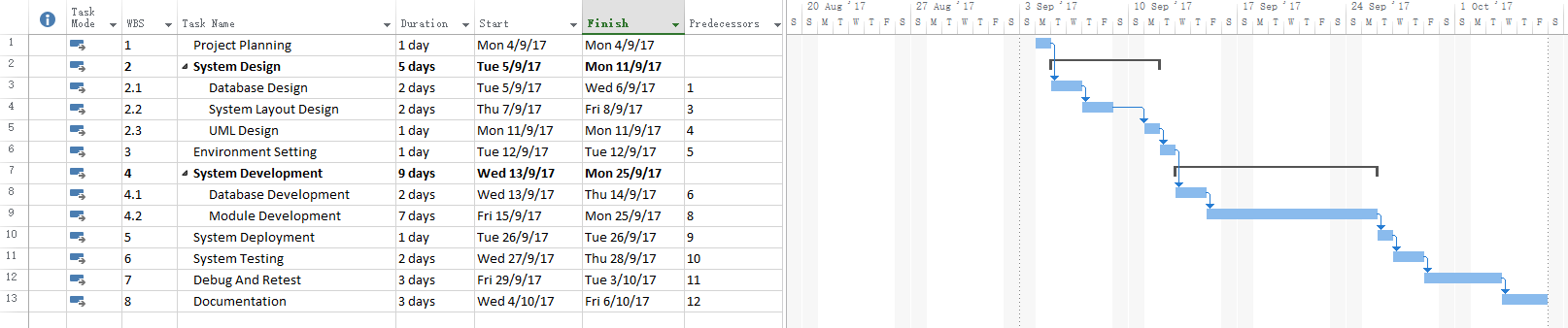
# Introduction

Online shopping and other online services are becoming more demand in nowadays. Smartphones and smart gadgets are improving the quality of life of people. Customers always want to able to perform something quickly including booking things and buying things.

The Ukraine International Airlines has an idea to look for new market, they want to set up a website for the customer to able to book tickets and view their booking online. But they are having some performance issue and security issues from the past.

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. Azure was also very compatible with open source software, which didn’t surprise Prudnikov.

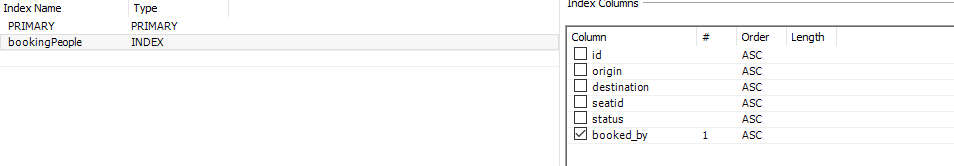
# Project Plan



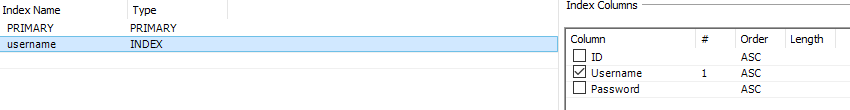
# Design

## Cloud Design Patterns

Database table indexing is used in the database for improving the database performance while retrieving data from the database.

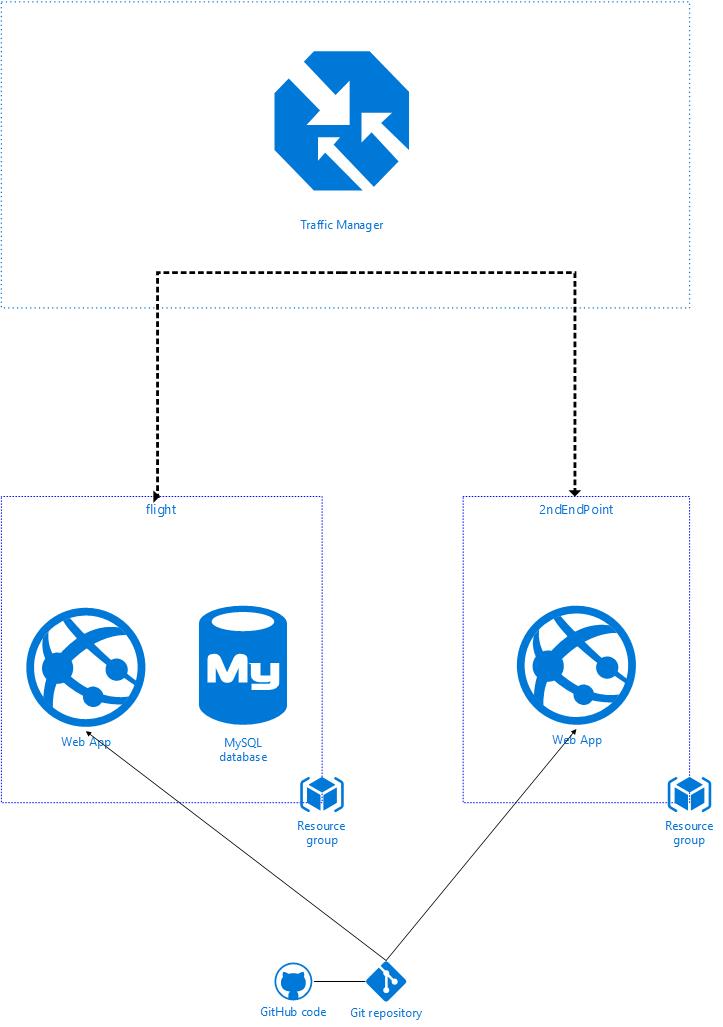


Booked\_by column in the booking table was index for searching purpose because there will be few thousands or more records in the database in the future. By implementing the index for the table is able to reduce the logical search to look for the correct records instead of run through all records of the table to find the record needed.



The username of the user table also indexed for the performance of the database. When performing login checking, the system is able to search the correct username in shorter time hence enhancing the performance of the system.

## Architectural Design



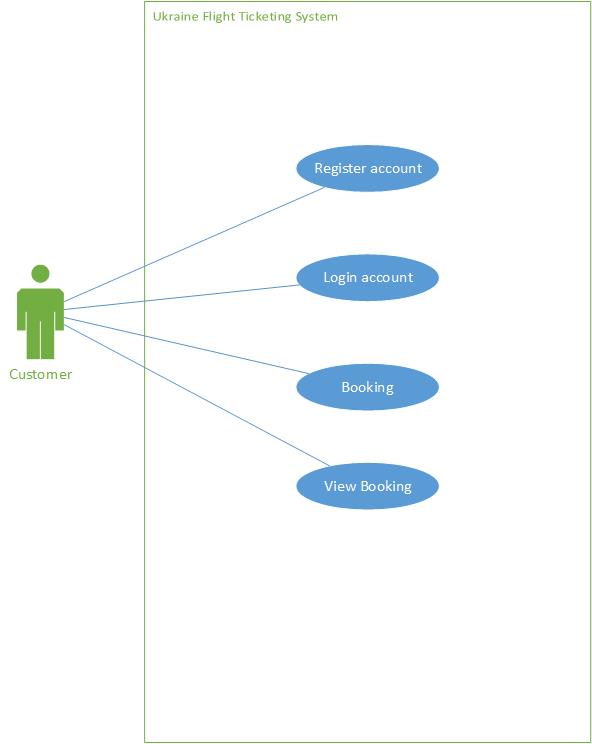
## Design Considerations

There are 2 endpoints deployed in South East Asia and East US. The online ticketing system would like to have as many end points as well but the budget given for hosting the system is only RM 150, so considering the budget we have, we deploy one of the endpoint in East US to cover western countries and one in South East Asia to cover Asia countries.

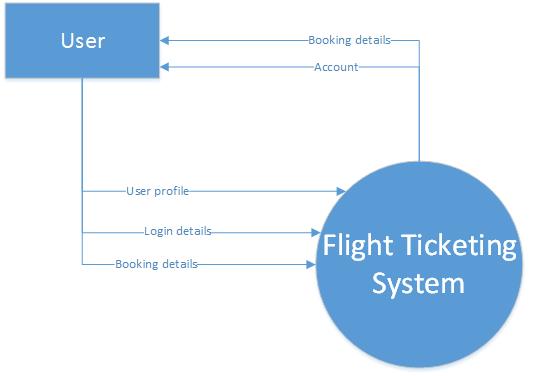
Traffic manager also applied for redirect customer who quite far away from these 2 endpoints based on the performance of the endpoint which one would give the customer the best experience.

## Modelling

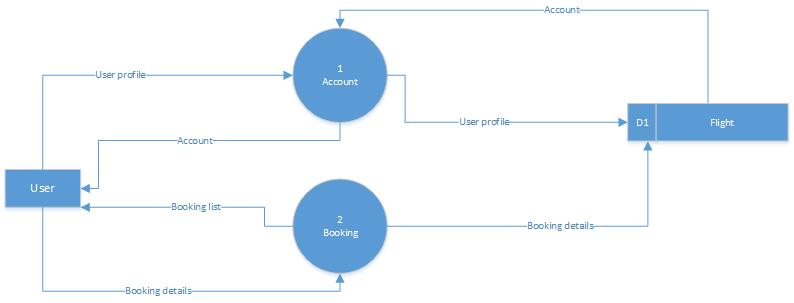
### Use Case Diagram



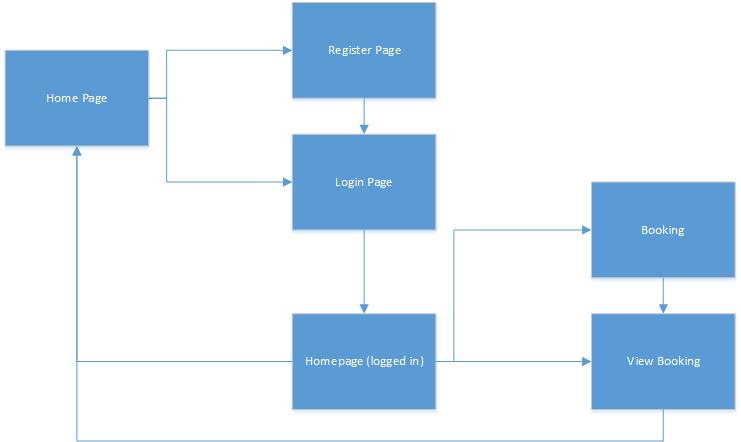
### Context Diagram



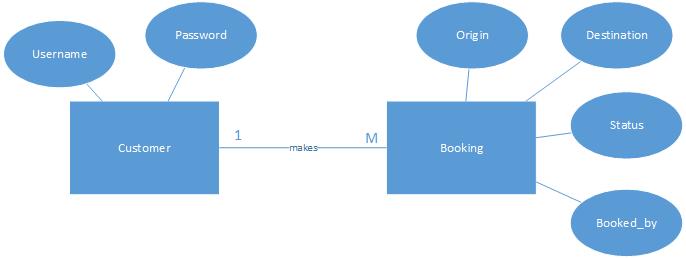
### DFD level 1



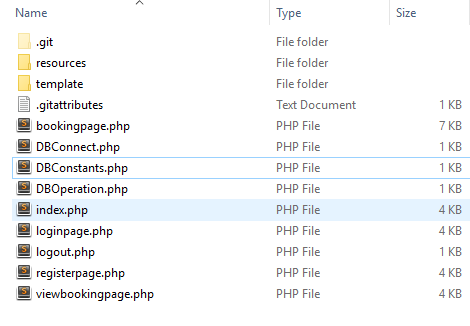
### Page Flow Chart



### Entity Relationship Diagram



# Implementation



The system is using the MVC model where the UI, controllers and the functions are separated. Programming languages used in this system is php, javascript and CSS. CSS is used to decorate the website and the javascript and php are to control the backend of the system. Resources contains all the js files, css files and images of the system. DBConnect, DBConstants and DBOperation are the classes made to connect to the database. DBConstant stores the server address, username, password and database name. DBConnect uses DBConstants’s variables to connect to database. DBOperation uses the connection bridge to create operations like select statement or insert statement into the database.

The system is tested on local then upload to the github repositories. After uploaded to the github repositories then the system is deployed in 2 end points. One is South East Asia and the other is East US. After deployed the system then the system is equipped with traffic manager to manage the traffic of the access of user. Traffic manager is managing the traffic by the performance of the system. Performance testing are done in the azure service to analyse the performance of the system.

Database chosen for the system is MySQL database. MySQL has highest security among all the databases and it provides the most reliable performance to the system. MySQL has high performance also because it has a distinct storage-engine framework that facilitates system administrators to configure the database for a flawless performance. One of the best part of MySQL is MySQL is open source, plenty of system are using MySQL as their database, although in Azure it need subscription to use the MySQL but it is still worth the amount of money paid.

# Test Plan

## Unit testing

### Pages Navigation

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case Name | Procedure | Expected Result |
| 1 | Homepage navigation | 1. Press Home on the header | Display homepage |
| 2 | Loginpage navigation | 1. Press Login on the header | Display Login page |
| 3 | Registerpage navigation | 1. Press Register on the header | Display register page |
| 4 | Booking page navigation | 1. Press Login on the header  2. Login with a valid account  3. Press Booking on the header | Display booking page |
| 5 | View booking page navigation | 1. Press Login on the header  2. Login with a valid account  3. Press View Booking on the header | Display view booking page |

### Login page

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case Name | Procedure | Expected Result |
| 6 | Login with incorrect account | 1. Press Login on the header  2. Login with an invalid account | Login page display incorrect username and password |
| 7 | Login with correct account | 1. Press Login on the header  2. Login with a valid account | Login page now redirect to home page with booking and view booking on the header |

### Register page

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case Name | Procedure | Expected Result |
| 8 | Press register without filling up form | 1. Press register on the header  2. Press register button | Register page display “Please fill up the form” |
| 9 | Press register with the form filled up | 1. Press register on the header  2. Fill up the form  3. Press register button | Register page display “Registered successfully” |

### Booking page

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case Name | Procedure | Expected Result |
| 10 | Seat map | 1. Press Login on the header  2. Login with a valid account  3. Press booking on the header  4. Press on any grid on the seat map | Text field below seat id shows the id pressed on the seat map |
| 11 | Press book without filling up form | 1. Press Login on the header  2. Login with a valid account  3. Press booking on the header | Booking page display “Please fill up the form” |
| 12 | Press book with the form filled up | 1. Press Login on the header  2. Login with a valid account  3. Press booking on the header  4. Press on any grid on the seat map  5. Fill up the form | Booking page display “Booked successfully” |

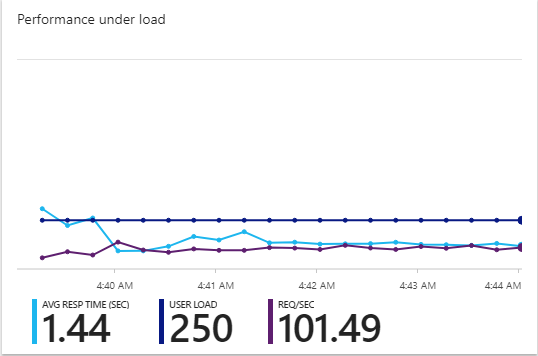
### View booking page

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case Name | Procedure | Expected Result |
| 13 | View booking | 1. Execute test case 12  2. Press view booking on the header | Booking details show on the view booking page |

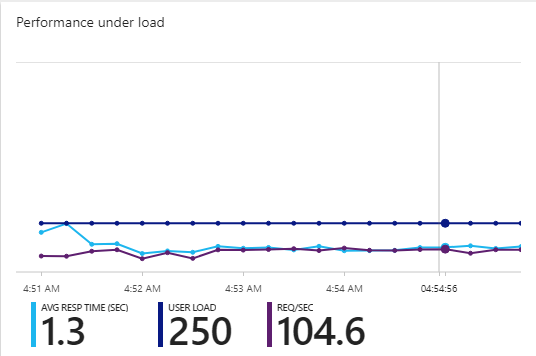
### Logout

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Test Case Name | Procedure | Expected Result |
| 14 | Logout | 1. Press Login on the header  2. Login with a valid account  3. Press logout | Redirect to homepage without booking and view booking on the header |

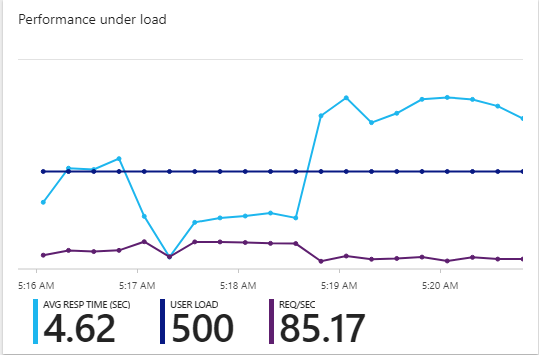
## Performance Test



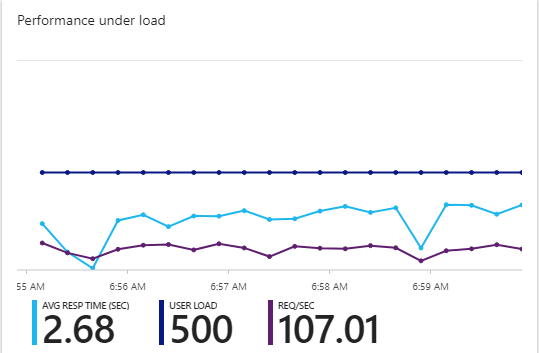
1st performance test with 250 user loads.



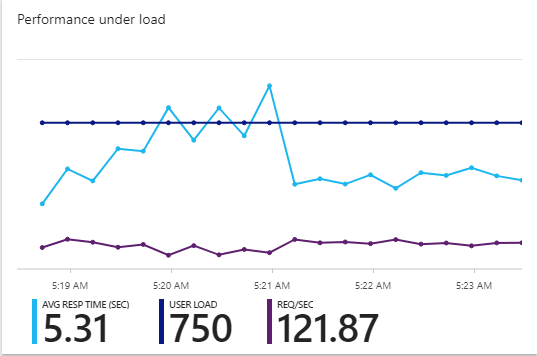
2nd performance test with same setting as 1st performance test. As the result can see 2nd test performance is better with average response time of 1.3 sec.



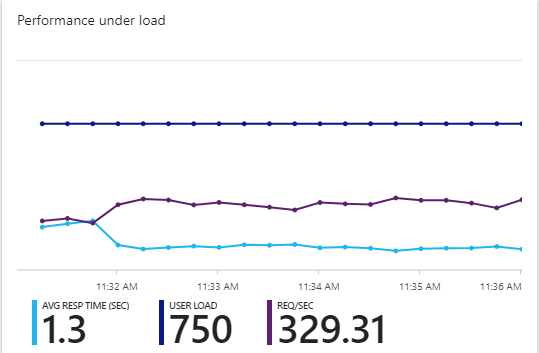
3rd performance test with 500 user loads.



4th performance test with 500 user loads but with traffic manager. The average response time is lower than without traffic manager.



5th performance test with 750 user loads.



6th performance test with 750 user loads and with traffic manager. Looks like it is faster than 4th performance test, perhaps the traffic during the time when running the test wasn’t too tight.

# Conclusion

Cloud services provides a lot of convenient to the developers. They can host their system in the cloud and manage their system in the cloud. There are a lot cloud service providers that provides many kinds of service to the developers. Maintenance cost is one of the biggest issue that can solve by having cloud hosted system, the maintenance fee of the hardware can be deducted from the outcome of developing the system.

The company who wish to deploy their system and database on cloud might have the data exposed to third party. Then it comes to the balance of either which one is more profitable, reduce the maintenance cost of the system or private data of company exposed to the third party.

# References

CONTRIBUTOR. (2017, July 26). *8 Advantages of Using MySQL*. Retrieved from Dev Ops: https://devops.com/8-advantages-using-mysql/

Github. (2016, April 6). *Github*. Retrieved from Github: https://guides.github.com/activities/hello-world/

Microsoft. (2016, Dec 2). *Azure App Service plans in-depth overview*. Retrieved from Microsoft Azure: https://docs.microsoft.com/en-us/azure/app-service/azure-web-sites-web-hosting-plans-in-depth-overview

Microsoft. (2017, August 9). *Cloud Design Patterns*. Retrieved from Microsoft Azure: https://docs.microsoft.com/en-us/azure/architecture/patterns/

MIcrosoft. (2017, August 9). *Icons and Diagrams*. Retrieved from Microsoft Azure: https://docs.microsoft.com/en-us/azure/architecture/resources/diagrams

Microsoft. (2017, June 6). *Scale instance count manually or automatically*. Retrieved from Microsoft Azure: https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/insights-how-to-scale?toc=%2fazure%2fapp-service-web%2ftoc.json#scaling-based-on-a-pre-set-metric

Posey, B. (2017, July 25). *First Look: Azure Web Application Firewall*. Retrieved from Redmond: https://redmondmag.com/Articles/2017/08/01/Web-Application-Firewall.aspx?Page=1

W3school. (2017). *PHP 5 Tutorial*. Retrieved from w3school: https://www.w3schools.com/php/