**Student (s) Number as per your student card:**

**\_\_\_10370308\_\_**

**Course Title:** Information Systems with Computing (January 2018)

**Lecturer Name:** Paul Laird

**Module/Subject Title:** Enterprise Information Systems

**Assignment Title:** Flight Reservations and Booking using Web Service and Oauth

1. **Introduction**

Breezy Travels is an online application for flight booking and reservations. The project has been taken in order to solve problems with traveling and filling out user information if interested in making bookings or reservations. Breezy Travels looks at the basic travel needs of people moving from one location to another either for businesses, leisure, tourism or whatsoever. In this project work which focuses on how Enterprise Information Systems are properly utilized, we look at using a web service to generate data that would be requested by the user and Oauth to populate information requested from the user by the system.

The web service would make it easy to get requested information from one location and distribute same and accurate information at any time requested.

* 1. **Definitions, Acronyms and Abbreviations**

**Web Service**

The term Web services describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDIopen standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Used primarily as a means for businesses to communicate with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall.

Unlike traditional client/server models, such as a Web server/Web page system, Web services do not provide the user with a GUI. Web services instead share business logic, data and processes through a programmatic interface across a network. The applications interface, not the users. Developers can then add the Web service to a GUI (such as a Web page or an executable program) to offer specific functionality to users.

1. **Technologies Used**

In order to achieve the set objective, various technologies were used all together, they would be listed in this section. The following are the core technologies used in developing the web application:

* 1. **Travel APIs**

For booking the best global flight, hotel and car rental deals, we used the allmyles api to fetch realtime flight data.

API Endpoint: https://dev.allmyles.com/v2.0/flights **Framework Use: ASP.Net core**

ASP.NET Core is a lean and composable framework for building web and cloud applications. ASP.NET Core is fully open source and can run on Windows, Mac, and Linux.

ASP.Net Core was an important tool used in the development of this application although the process wasn’t easy since there wasn’t so much knowledge in this area, but there were lots of research to get the app up and running. ASP.Net core is available on visual studio for majorly c-sharp developers.

* 1. **Entity Framework (For Accessing the Database)**

Entity Framework is an object-relational mapper (O/RM) that enables .NET developers to work with a database using .NET objects. It eliminates the need for most of the data-access code that developers usually need to write.

The database concept was adopted in this project so that user information can be stored either when they register normally or using the provided Oauth. In the case of using the provided Oauth service, the information gotten from the authenticating app, is saved into the database. Using the entity framework resource available on asp.net, email addresses and chosen passwords are collected. Information collected includes names or passenger and so on.

Entity Framework (EF) is an object-relational mapper that enables .NET developers to work with relational data using domain-specific objects. It eliminates the need for most of the data-access code that developers usually need to write. Entity Framework allows you to create a model by writing code or using boxes and lines in the EF Designer. Both of these approaches can be used to target an existing database or create a new database.

* 1. **ASP.Net Identity**

ASP.NET Core Identity is a membership system that adds login functionality to ASP.NET Core apps. Users can create an account with the login information stored in Identity or they can use an external login provider. Supported external login providers include Facebook, Google, Microsoft Account, and Twitter.

Identity can be configured using a SQL Server database to store user names, passwords, and profile data. Alternatively, another persistent store can be used, for example, Azure Table Storage.

* 1. **Authentication: Oauth**

The OAuth 2.0 authorization framework enables third-party applications to obtain limited access to a web service. It is an open protocol to allow secure authorization in a simple and standard method from web, mobile and desktop applications.

OAuth 2 is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service, such as Facebook, GitHub, and DigitalOcean. It works by delegating user authentication to the service that hosts the user account, and authorizing third-party applications to access the user account. OAuth 2 provides authorization flows for web and desktop applications, and mobile devices.

This informational guide is geared towards application developers, and provides an overview of OAuth 2 roles, authorization grant types, use cases, and flows.



In our application we used Facebook Web Service for Authentication users to our application.

Google and add stuff about facebook Oauth.

Here is a more detailed explanation of the steps in the diagram:

1. The *application* requests authorization to access service resources from the *user*
2. If the *user* authorized the request, the *application* receives an authorization grant
3. The *application* requests an access token from the *authorization server* (API) by presenting authentication of its own identity, and the authorization grant
4. If the application identity is authenticated and the authorization grant is valid, the *authorization server* (API) issues an access token to the application. Authorization is complete.
5. The *application* requests the resource from the *resource server* (API) and presents the access token for authentication
6. If the access token is valid, the *resource server* (API) serves the resource to the *application*
   1. **IIS Server**

IIS is a web server that runs on the Microsoft .NET platform on the Windows OS. While it’s possible to run IIS on Linux and Macs using Mono, it’s not recommended and will likely be unstable. (There are other options, which I’ll present later). It’s versatile and stable, and it’s been widely used in production for many years. Version 10 is the most current. Once it’s installed you’ll see this welcome page in your browser.

**Internet Information Services** (**IIS**, formerly **Internet Information Server**) is an extensible web server created by Microsoft for use with the Windows NT family. IIS supports HTTP, HTTP/2, HTTPS, FTP, FTPS, SMTP and NNTP. It has been an integral part of the Windows NT family since Windows NT 4.0, though it may be absent from some editions (e.g. Windows XP Home edition), and is not active by default.

1. **The Design**

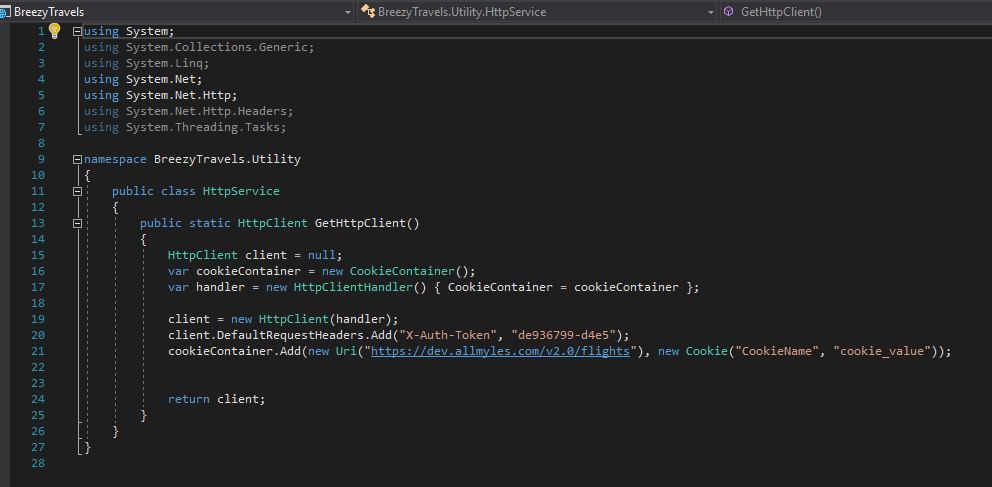
ASP.Net provides developers with the Model Views and Controllers Concept which enables the developer develop in a well-structured manner. This app was developed using the MVC framework.

The models were used to handle Data related requests and were used to also parse to the view.

The Controllers were used to handle requests between the models and the views resulting in a perfect connection between the period when the user makes a request and the period when data is gotten from the database.

The views were used to handle the User interface and the views were populated in Razor which is an available asp.net resource.

For the web service, the HttpService and HttpClient resource available on asp.net was used.



From the above image we see the Token as provided by Allmyles and the Uri which shows where the connection is coming from.

1. **The Artefact**

* The application focuses on fetching its data from the web service provider called AllMyles. When the user makes the search or searches for available flights and dates, data is gotten from the API incorporated in the code and then results are displayed to the user. The airline name, flight departure date and time, number of seats remaining are all displayed for the user to see.
* Also the Oauth function helps the user automatically enter his /her details after authentication from facebook.
* The user simply searches for the available flights and then clicks on book, which then gives access to the Oauth page.

1. **Challenges**

Going through the process of creating a full app as an individual in a short period of time wasn’t an easy task although I was able to search online for resources in order to come to a good end.

Also one major discovery was that the api used to populate data was not giving correct or accurate data which can be a great limitation. There were lots of errors discovered from the API but in all, the purpose was met as per the requirements of this assessment.

1. **RESOURCES**

<https://www.webopedia.com/TERM/W/Web_Services.html>

<https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>

<https://stackify.com/iis-web-server/>

<https://msdn.microsoft.com/en-us/library/aa937723(v=vs.113).aspx>

1. **APPENDIX**

