Content

[**1.** **Purpose of Application** 2](#_Toc359692090)

[**2.** **System Architecture** 2](#_Toc359692091)

[**2.1 Main Components** 2](#_Toc359692092)

[**2.2 Distributed Components** 2](#_Toc359692093)

[**2.3** **Components using cloud service** 3](#_Toc359692094)

[**2.4** **System Components Diagram** 3](#_Toc359692095)

[**3. Class Diagrams** 3](#_Toc359692096)

[**3.1 Database Schemas** 3](#_Toc359692097)

[**3.2 Classes** 4](#_Toc359692098)

[**3. 3 Program Implementations** 5](#_Toc359692099)

[**4.** **Test Plan** 5](#_Toc359692100)

[**4.1** **Platform Details** 5](#_Toc359692101)

[**4.2** **Login Details** 5](#_Toc359692102)

[**4.3** **How Operations Performed** 6](#_Toc359692103)

[**5.** **Known Issues** 9](#_Toc359692104)

1. **Purpose of Application**

The application developed for this assignment is airline reservation system. This airline reservation system can be used by normal clients for booking flights, and also can be used by system administrator to manage amounts of flights, airlines and various seat configurations.

Moreover, this system is also one of the best examples to demonstrate windows azure cloud service. In my project, I mainly demonstrate security features by using login form and password decryption, and also demonstrate the fault tolerance in case of database service go offline.

1. **System Architecture**

**2.1 Main Components**

In my airline reservation system, the necessary component required to run the system should be the database system, database service, and a server host. These three components mentioned above are made up of the server-side service. On the other hand, as for the client-side representation, I apply WPF windows as the front-side presentation for the clients. Details about these components are shown below:

* **Database System:** SQL server is used as the underlying database system, which contains the tables (users, airlines, flights, seats, orders table) and views (viewflights and vieworders). This database system located in the windows azure cloud, which is used by the WCF.
* **Database Service:** This service is defined in WCF, which including the library and actual service application. Moreover, this service is also divided into two parts (like database service and database backup service) for ease of failure handling. In addition, class diagram and business logic (like sql CRUD operation and other functions) are also implemented in this service node. This service is also uploaded to the windows azure, which is accessed by the server host.
* **Server Host:** This host manages all the server-side service designed in my project. What is more, it binds the WCF service to the proper address, so that the client can connect to the server and perform necessary tasks. The job of this routing server is to redirect client messages to a suitable end point by writing app.config file. (in my project, It points to the cloud service available at the windows azure)
* **WPF Client:** These are multiple WPF windows which can navigate with each other, which can be used by client as well as the administrator.

**2.2 Distributed Components**

The distributed Components of my system should be the WCF service part and the database system part. The reason is that WCF service is now available on the windows azure, so it can be accessed by any clients with specifying the right site URL.

As for the database system, it can be distributed component because the connection string of the database system points to server URL provided by the windows azure. So anyone at anyplace can make use of the database with this connection.

* 1. **Components using cloud service**

The components using the cloud service are the database system and the WCF service. The database system uses the Azure SQL database instead of the local database by specifying the connection link provided by the windows Azure. As for the WCF service, it is uploaded as the cloud service, and it is used by the routing server in the project.

* 1. **System Components Diagram**

Admin Interface WPF Application

Client Interface WPF Application

Airline Reservation system

Client-Side

Host

Database Server

Database Service

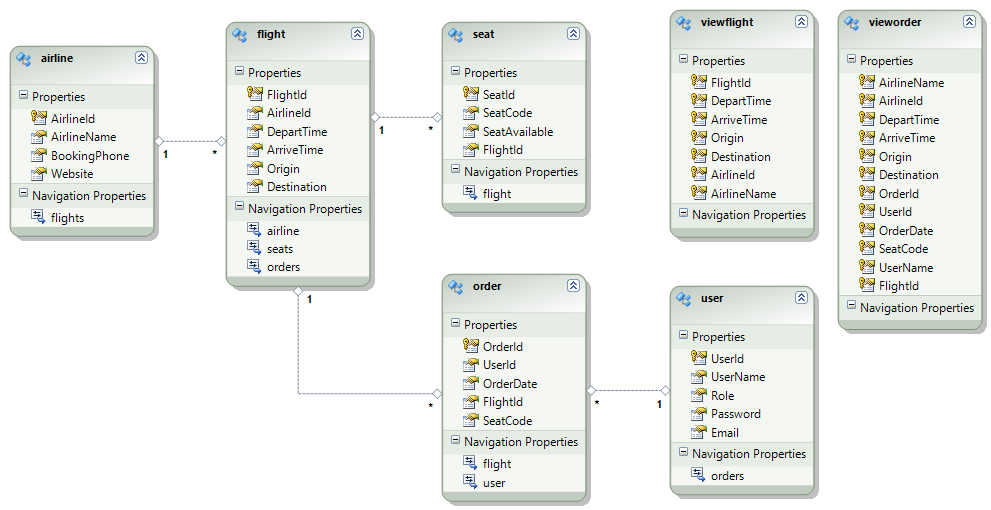
Server Host

Server-Side

Connect

**3. Class Diagrams**

**3.1 Database Schemas**

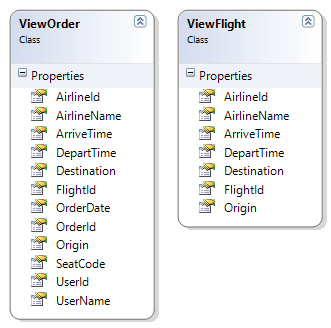


**3.2 Classes**

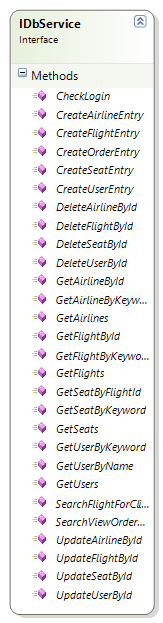
**Classes Defined For Database Table Object**



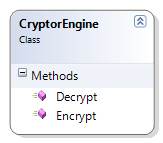
**Class Defined For Database View Object**



**Class Defined For Service Interface (implemented by wcf service)**



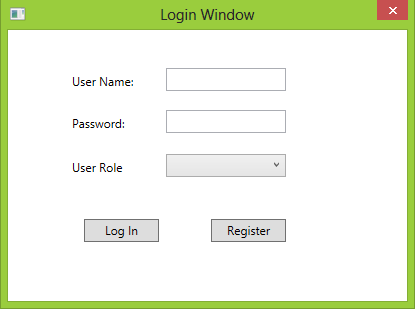
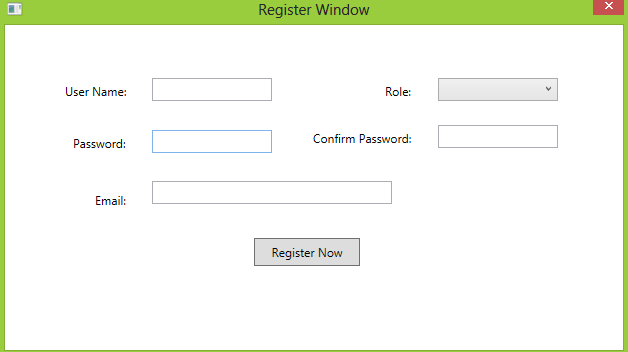
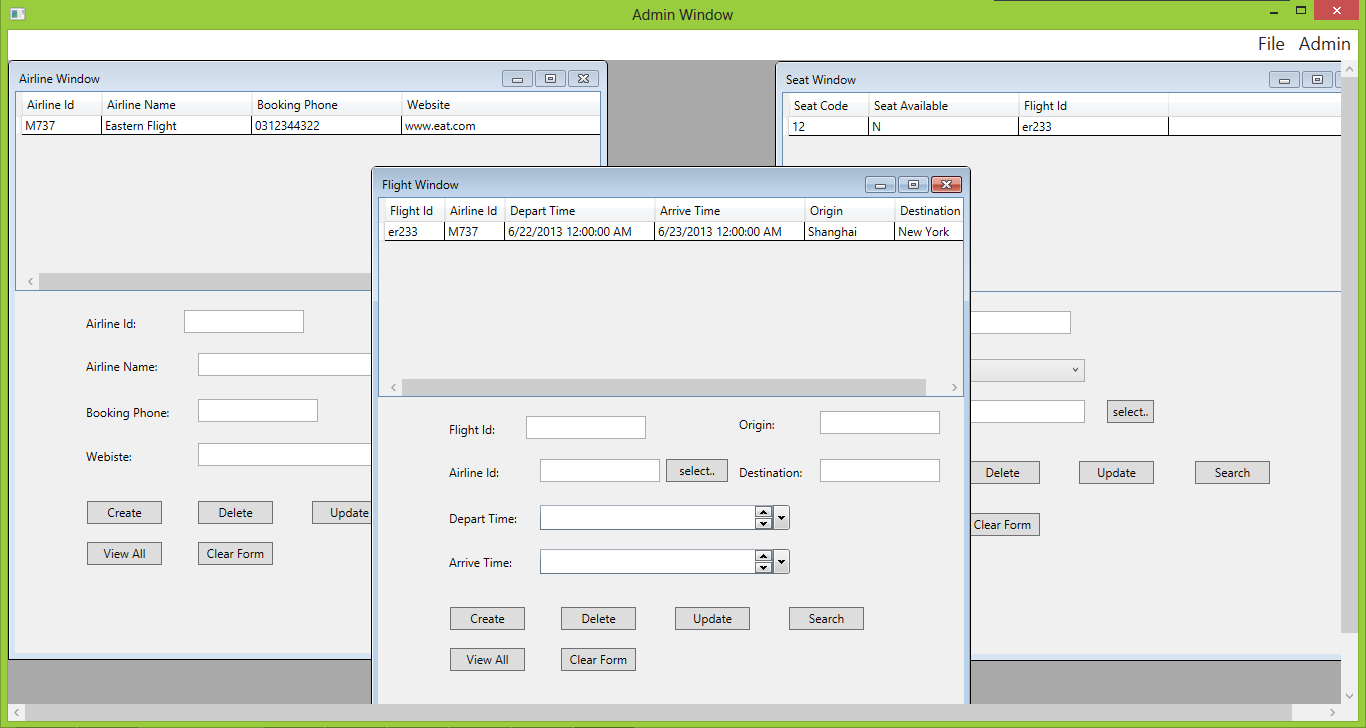
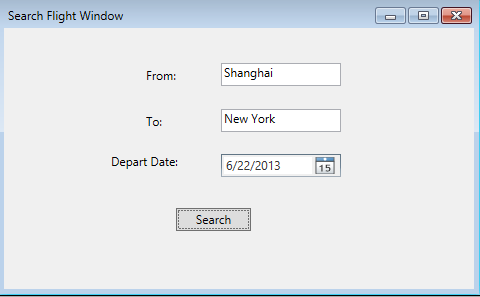
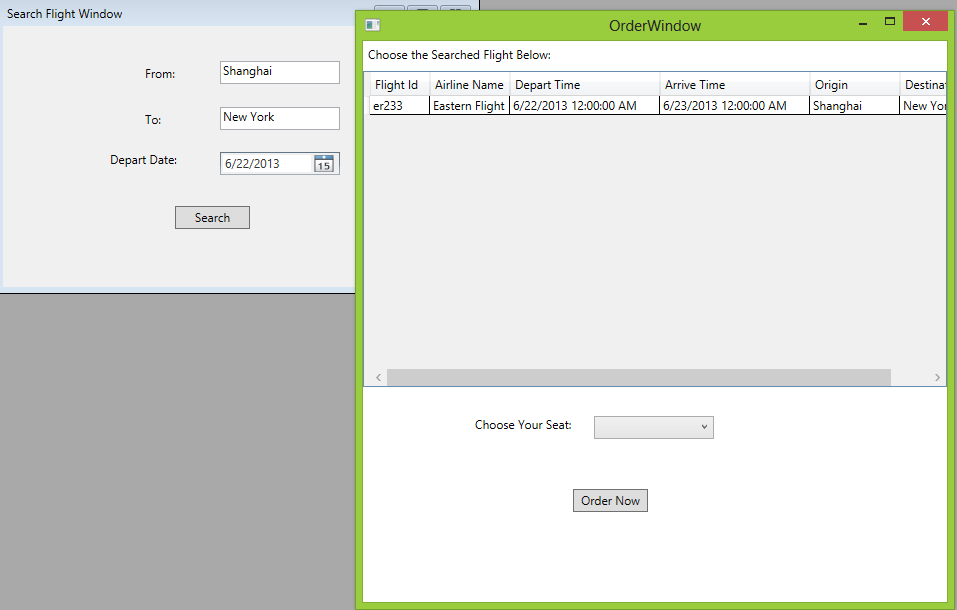
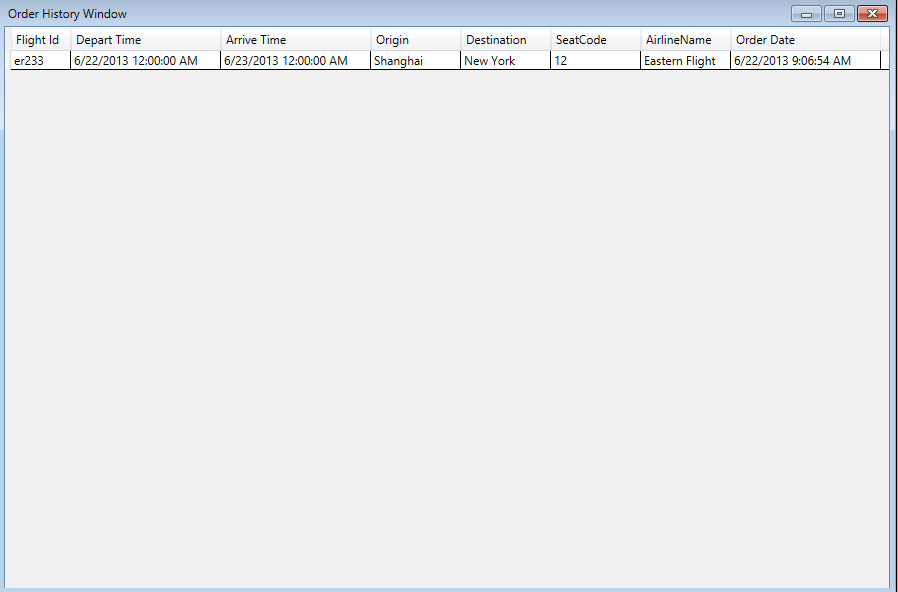
**Other Class for Password Encrypt and Decrypt**



**3. 3 Program Implementations**

To implement the program, it is necessary to start the router server. As a result, the client interface (WPF) can visit the WCF service defined in the application. WCF service make use of the database resource, which including the CRUD operations.

1. **Test Plan**
   1. **Platform Details**

* Visual studio version: VS 2010
* Toolkit and packages: VS 2010 Service Pack 1,Windows Azure SDK for .NET (VS 2010 SP1) 1.8.1, Windows Azure Libraries for .NET 1.8
* Instruction: just install the package and VS2010 above
  1. **Login Details**
* Login details: admin username:123 password:123
  1. **How Operations Performed**
* set the WpfClient as the start project
* run the cmd as system administrator and enter the following command: *netsh http add urlacl url=http://+:8200/AirportSolution/service user=<user>*
* locate the folder(*RoutingExample\RoutingService\bin\Debug*), run the RoutingService.exe
* Run the solution in VS2010
* After you run the program, you will be directed to the login interface (combined the admin login and client login). If you are the new user, you can register as well.  
    
  
* When you login as administrator, you can go to the Admin interface, where you can do CRUD operations on the table existing in the database. The system will validate user input as well.  
  
* When you login as a client, you will go to the client window. You can order the flight, and view the order history here. You can search for the flight by typing its origin and destination and departing date. Then you can choose your seat number, and click Order will generate an order. Order history can be viewed through clicking the menu item.  
    
    
  

1. **Known Issues**

When the system grows larger, potential issues that may occur are:

* System Latency: Take the login function of the system for example. As the system does not have concurrency function, so the user cannot login concurrently. In order to connect to the system, the user needs to wait until other user finishes the connection.
* System Overload: if there are too many clients connecting to the server, the workload will be too high, which may result in a large degree of service degradation. Even worse, the server may breakdown finally.
* Compatibility Issue: Because the application is developed in the windows development environment, some clients who use Linux, Mac may meet troubles when starting the project.

Other potential issues related to the cloud architecture are:

* Data security: cloud computing provides a complex environment, which is hard to ensure the user data in secure. Issue like data integrity, privacy and data location will place bad effect on the trust of cloud computing.
* Performance: With the number of users growing up, the workload may greatly slow down the cloud service.
* Compatibility: Windows Azure is based on the Mircrosoft’s technology, which may not be applied on other platforms.

Bugs:

* Bugs related to update table: Take the airline table in database for example. Because airline Id is selected as the primary key, it is not permitted to update the airline Id by the administrator.
* Bugs related to create record: Take the user table in database for example. Because user Id is selected as the primary key, the user name may be repeated which is not consistent real scenario.
* Bugs related to delete record: Take the airline table and flight table for example. Because the airline id in flight table needs to reference the airline id in airline id, it will cause error if the user wants to delete the airline record which is referenced by flight table.
* Bugs related to user role: It is necessary to build a system which need to verify whether the user has right to register as Admin.