



# **Dharmsinh Desai University, Nadiad**

Faculty of Technology  
Department of Computer Engineering

B. Tech CE Semester – VI

Subject: Service Oriented Computing

Project Title:

## **Car-Pooling Application**

By:

1) Rutvik Kathrecha

CE-058

17CEUBS048

2) Samyak Mehta

CE-067

17CEUON035

Guided by: Prof. Ankit P. Vaishnav



# **Dharmsinh Desai University, Nadiad**

Faculty of Technology

Department of Computer Engineering

## **CERTIFICATE**

This is to certify that the Service Oriented Computing project entitled “**Car-Pooling Application**” is a bona fide report of the work carried out by

- 1) **Rutvik Kathrecha, Roll No: CE058, Student ID No: 17CEUBS048**
- 2) **Samyak Mehta, Roll No: CE067, Student ID No: 17CEUON035**

Of Department of Computer Engineering, Semester VI, academic year 2019-20, under the guidance and supervision for the subject Service Oriented Computing.

(Project Guide)

**Prof. Ankit P. Vaishnav**  
Assistant Professor  
Department of Computer  
Engineering,  
Faculty of Technology,  
Dharmsinh Desai University,  
Nadiad

(HOD)

**Dr. C. K. Bhensdadia**  
Head of Department  
Department of Computer  
Engineering,  
Faculty of Technology,  
Dharmsinh Desai University,  
Nadiad

## Table of Contents

1. Abstract.....	4
2. Introduction.....	5
2.1 Brief Introduction.....	5
2.2 Technology and Tool Used.....	5
3. Software Requirement Specifications.....	6
4. Design.....	7
4.1 Use Case Diagram.....	7
4.2 Class Diagram.....	8
4.3 Sequence Diagram.....	9
4.4 Activity Diagram.....	10
4.5 E-R Diagram .....	11
4.6 Data Dictionary .....	12
5. Implementation Details .....	13
5.1 Modules .....	13
6. Testing .....	14
6.1 Testing Method.....	14
6.2 Test Cases .....	14
7. Screen-shots of the System .....	15
8. Conclusion .....	18
9. Limitations and Future Extensions of System .....	19
10. Bibliography .....	20

## Abstract

In today's world everyone wants some kind of sharing in life so that they can achieve better life without paying high cost. One of the sharable things is transportation. Daily lots of people up-down from cities. Some have their personal vehicle and some use local transport. People having personal vehicle fills they are costing much and people using local transport fills they are tired much. So, let's connect both and come to the idea of carpooling.

## Introduction

In today's world everyone has value of two things: time and money. Everybody wants somehow to save their time and money. Carpooling is the project where these both aspects will be satisfied. The person having own vehicle can save money by sharing vehicle with others who are on-route. The other person who is using local transport can take the service of these type of people to save time and take easier journey at same price.

In our application we have built one service for admin site which provides actions like view all cars, add, update or delete car details. So, basically it provides the CRUD operation on car object. We have used service-oriented approach to build this application.

### **Technology used:**

- .NET framework
- WCF Service
- Windows Forms

### **Platform used:**

- Visual Studio 2019

# Software Requirement Specifications

## **System Functional Requirements**

### **R.1 Admin**

#### **R.1.1 Login**

Description: If you already have an account, login here.

Input: Username, password.

Output: Successfully logged in.

Processing: Admin will enter username and password to login, if already registered.

#### **R.1.2 Manage Cars**

Description: Admin can perform all the CRUD operation on cars.

##### **R.2.4.1 View Cars**

Input: User selection

Output: Display Cars details.

##### **R.2.4.2 Add Car**

Input: Car details.

Output: Car successfully added.

##### **R.2.4.3 Update Car**

Input: Car details.

Output: Car successfully updated.

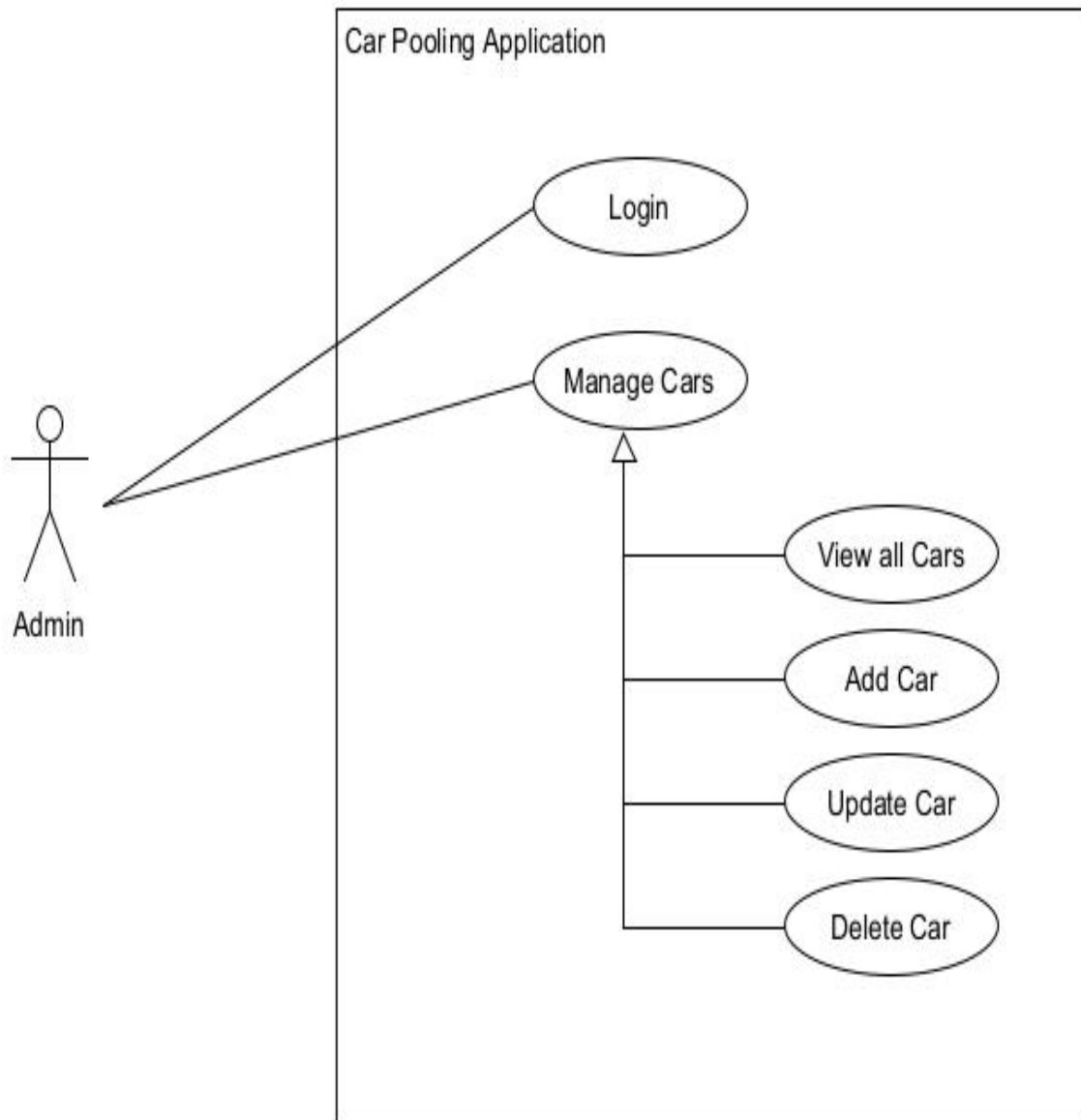
##### **R.2.4.4 Remove Car**

Input: Car details.

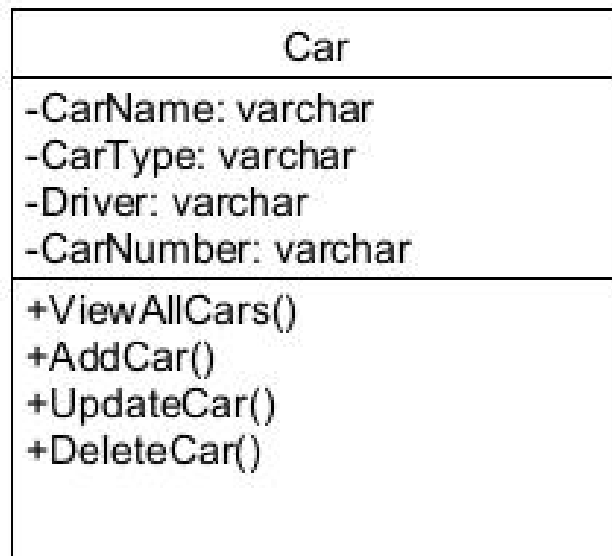
Output: Car successfully removed.

# Design

## 1) Use Case Diagram

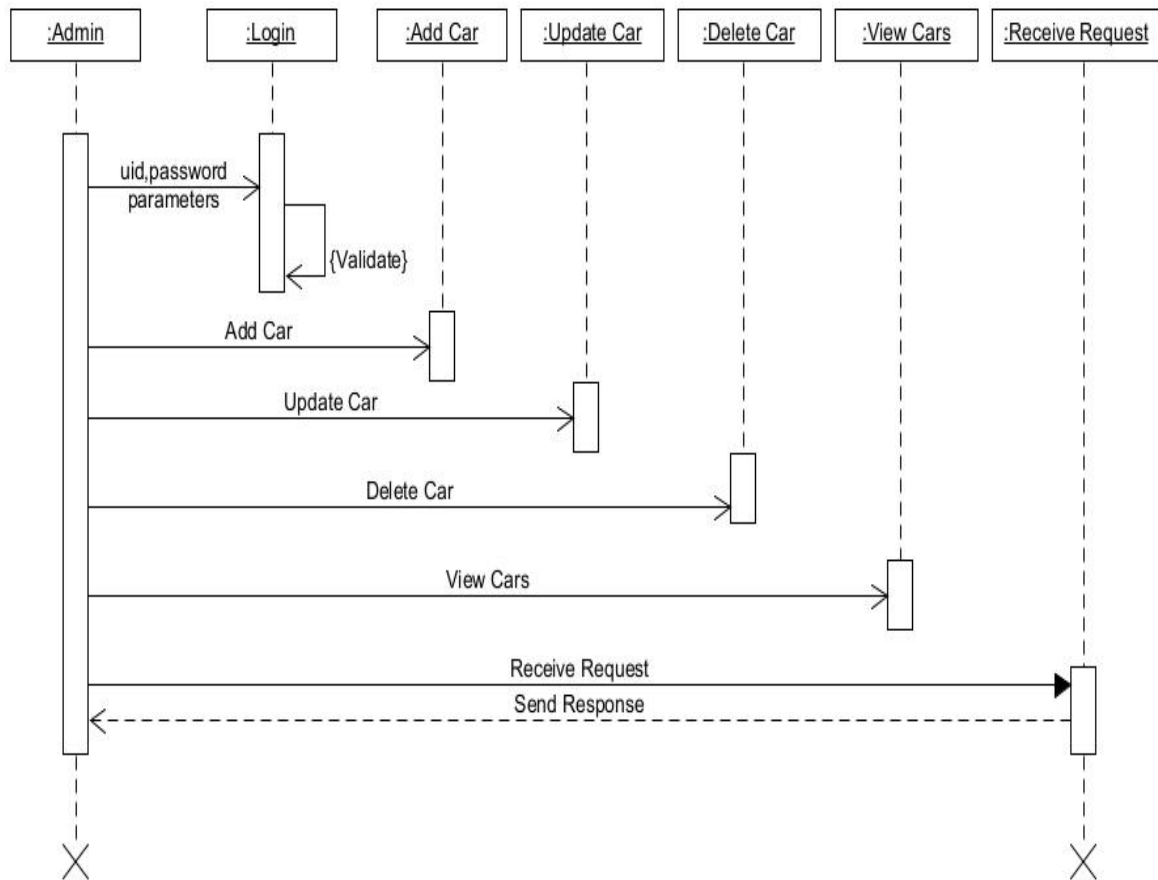


## 2) Class Diagram

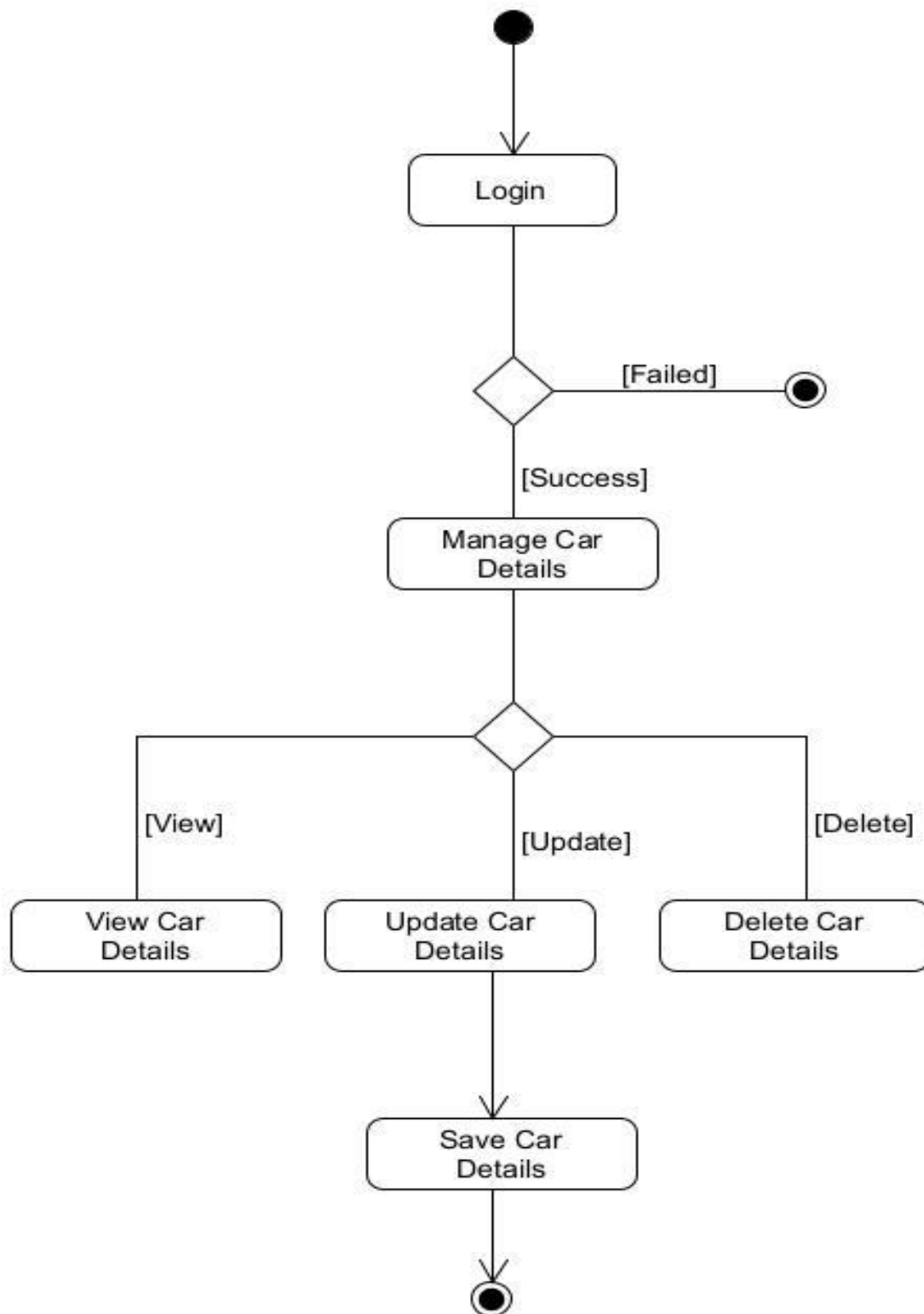




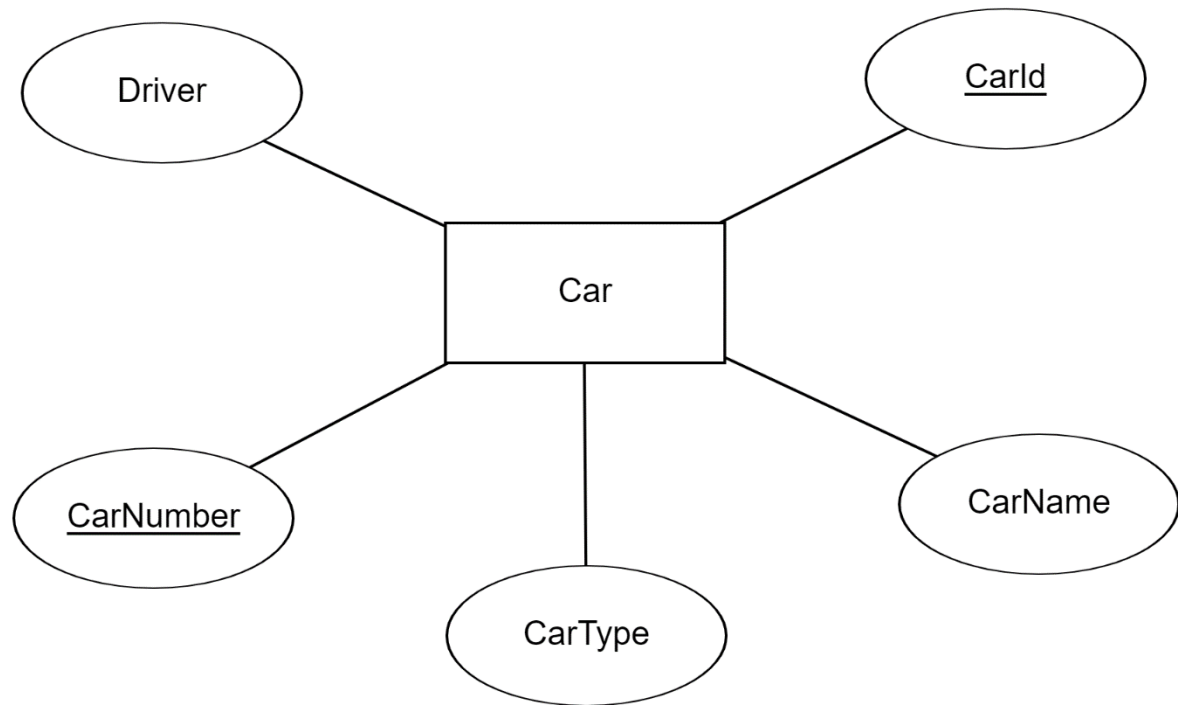
### 3) Sequence Diagram



## Activity Diagram



#### 4) E-R Diagram



### 5) Data Dictionary

Car							
Sr. No.	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table
1	CarId	Number	20	Yes	Yes	PK	
2	CarType	Varchar2	50	Yes	No		
3	CarName	Varchar2	50	Yes	No		
4	Driver	Varchar2	50	Yes	No		
5	Car_Number	Varchar2	50	Yes	Yes		

# Implementation Details

## Manage Car Details Module

Admin can add the car in the system. For that admin has to provide car details like Car id, Car type, Car name, Driver, and Car number. And if details are valid than data will be inserted.

If Admin wants to update the car details then he/she can write the Car Id and select that particular details and update the details.

The Admin can also Delete the Car details by writing the Id.

Also, The Admin can Get all the car details by selecting view all cars.

# Testing

We have performed black box testing for this system.

The screenshot shows a web application interface for managing cars. On the left, there are input fields for 'Id-' (value: 2), 'Car\_Type' (value: small), 'Car\_Name' (value: Nano), 'Driver' (value: root), and 'Car\_Number' (value: gj-16-h-2341). Below these fields are three buttons: 'Insert' (highlighted in blue), 'Update', and 'Delete'. At the top right of the main area are two buttons: 'Select' and 'Select All'. In the center is a table with the following data:

	Car_Name	Car_Number	Car_Type	Driver	Id
▶	city	GJ-16-k-4567	sedan	sam	2
	innova	gj06-h-6789	big	man	3
	xuv500	gj-16k-2341	big	sam	4

A modal dialog box is open in the bottom right corner with the title 'CarId already taken' and an 'OK' button.

- Here, Already Car Id with 2 is inserted, so when any other user tries to enter car with same Id then it will not be allowed so that each and every car will have unique car Id.

## Screen-shots

The screenshot displays a web application interface for managing car details. On the left, there is a form with the following fields and values:

- Id: 6
- Car\_Type: big
- Car\_Name: BMW
- Driver: Rutvik
- Car\_Number: GJ-03-AB-7777

Below the form are three buttons: Insert (highlighted in blue), Update, and Delete.

On the right, there is a DataGrid with the following columns: Car\_Name, Car\_Number, Car\_Type, Driver, and Id. The grid contains three rows of data:

Car_Name	Car_Number	Car_Type	Driver	Id
innova	gi06-h-6789	big	root	3
xuv500	gi-16k-2341	big	sam	4
Honda city	GJ16-K-2341	sedan	sam	5

At the top of the DataGrid are two buttons: Select and Select All. A modal dialog box is centered on the screen, displaying the message "Car Details Inserted Successfully" and an OK button.

- The above figure shows that car details are inserted successfully and in below figure it is clearly seen in the DataGridView that data with Car Id = 6 is inserted.

Id-

Car\_Type

Car\_Name

Driver

Car\_Number

Select

	Car_Name	Car_Number	Car_Type	Driver	Id
▶	innova	gj06-h-6789	big	root	3
	xuv500	gj-16k-2341	big	sam	4
	Honda city	GJ16-K-2341	sedan	sam	5
	BMW	GJ-03-AB-7777	big	Rutvik	6

×

Car details deleted successfully

- The Car Id has Value 4 and Message Box shows that Car Id is Deleted.

Id-

Car\_Type

Car\_Name

Driver

Car\_Number

Select

	Car_Name	Car_Number	Car_Type	Driver	Id
▶	Honda city	GJ16-K-2341	sedan	sam	5

×

Car details updated successfully

- Here Details of Car id 5 that is Car\_Name and Car\_Number is changed and updated.
- The result of Delete operation and Update Operation is as follows:



Id-

Car\_Type

Car\_Name

Driver

Car\_Number

Insert

Udate

Delete

Select

Select All

	Car_Name	Car_Number	Car_Type	Driver	Id
▶	innova	gl06-h-6789	big	root	3
	Swift Dezire	GJ16-K-0123	sedan	sam	5
	BMW	GJ-03-AB-7777	big	Rutvik	6

## Conclusion

In this car-pooling application we have successfully implemented CRUD operation on the car details. We implemented this using the WCF-Service which is another goal of the application. We have successfully implemented the Admin module for the application in which admin can View, Insert, Update, and Delete car details.

# Limitations and Future Extension

## Limitations

The system does not check the authenticity of the Driver whether the car is owned by him/her only. That work is done by admin manually.

## Future Extension

- To implement the client module for application.
- To implement the functionality for checking the authenticity of driver
- To provide functionality which supports Deletion of the Driver as well as Car details automatically on getting complaints about service provided or rude behaviour of the driver.

## Bibliography

- <https://docs.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf>