# Credit Scoring FinTech : The sample online credit application using WCF services and multi-platform client technology

**Kayun Chantarasathaporn[1*], Sudasawan Ngammongkolwong[2] and Kantikamaht Rattanaparinyanukool[1]**

[1] Department of Finance, Faculty of Business Administration, Huachiew Chalermprakiet University
Samutprakarn 10540

[2] Department of Multimedia Technology and Animation, Faculty of Science and Technology,
Southeast Bangkok College, Samutprakarn 10260

## Abstract

This article is related to overview and sample practical usage of financial technology or FinTech. It divides FinTech to 3 groups, enhancement of traditional financial services, innovative financial tools and services and algorithm related for executive support services. To have a real touch of FinTech, this article offers sample basic program of online credit application processing based on credit scoring concept. This program is 2-tier application. Both client and server software are written in C#. Server is responsible for business logic and is written based on WCF services which is the enhancement from open standard web services architecture. There are 2 platforms of sample client. One is written as windows application while another is created as web application. This article is useful for staff from both financial and information technology section.

**Keywords:** FinTech, Financial technology, Credit score, WCF services, Web application

**Corresponding author:** dr.kayun@gmail.com

## บทคัดย่อ

บทความนี้เกี่ยวกับภาพรวมและตัวอย่างภาคปฏิบัติของเทคโนโลยีทางการเงินหรือ FinTech เทคโนโลยีทางการเงินสามารถแบ่งได้เป็น 3 กลุ่ม คือ บริการที่เป็นการเพิ่มเติมธุรกรรมทางการเงินแบบเดิม บริการและเครื่องมือที่เป็นนวัตกรรมทางการเงิน และบริการที่เกี่ยวกับอัลกอริทึมเพื่อการสนับสนุนงานผู้บริหาร เพื่อจะได้สัมผัสกับเทคโนโลยีทางการเงินอย่างแท้จริง บทความนี้จึงนำเสนอตัวอย่างโปรแกรมขั้นพื้นฐานเกี่ยวกับการจัดการใบสมัครสินเชื่อออนไลน์ ซึ่งอิงกับหลักการเรื่องคะแนนสินเชื่อหรือเครดิตสกอร์ ตัวอย่างโปรแกรมเป็นแอพพลิเคชันแบบ 2 เทียร์ ทั้งซอฟต์แวร์ฝั่งไคลเอนต์และเซิร์ฟเวอร์เขียนขึ้นด้วยภาษา C# โปรแกรมเซิร์ฟเวอร์จะเกี่ยวกับตรรกะทางธุรกิจและเขียนขึ้นโดยอิงเทคโนโลยี WCF เซอร์วิส ซึ่งเป็นส่วนขยายมาจากสถาปัตยกรรมมาตรฐานเว็บเซอร์วิส โปรแกรมฝั่งไคลเอนต์มี 2 รูปแบบ คือ วินโดวส์แอพพลิเคชันและเว็บแอพพลิเคชัน บทความนี้เป็นประโยชน์ต่อกลุ่มบุคลากรทั้งด้านการเงินและด้านเทคโนโลยีสารสนเทศ

**คำสำคัญ:** FinTech เทคโนโลยีทางการเงิน คะแนนสินเชื่อ WCF เซอร์วิส เว็บแอพพลิเคชัน

### Introduction

Several years ago, FinTech might be considered as a buzzword [1] but now nobody can ignore it as its market value grows up real fast from 930 million dollars in 2008 to 12 billion dollars in 2015 [2]. There are many fields in FinTech [3], however, this article will mention just about the practical basic example of applying popular IT knowledge, WCF services, in helping automatic processing of consumer credit application based on the classic financial concept, credit score. Automatic online credit scoring can help lenders reduce cost roughly a thousand dollars per loan [4].

This article is going to talk about overview of FinTech, brief of credit score and sample basic usage of WCF services in automating accept/reject decision for consumer credit application. Server side code is written by using WCF services architecture while sample clients are shown in 2 platforms, windows and web applications. Source codes of both server and client sides are written in C# and the essential parts are listed as well.

### Big picture of FinTech

FinTech is abbreviated from Financial Technology. Wharton FinTech group gives the definition for it as “an economic industry composed of companies that use technology to make financial systems more efficient” [5]. FinTech uses the success of IT, computer network and telecommunication infrastructure to not only support existing financial related business but also create new financial tools and services. One of the main reasons that helps FinTech grow fast is the stability, enhancement and rather low price of the Internet.

Besides, regular computer internet connection, the spread of mobile internet usage via smart phone also pushes the online life. DazeInfo reported that 83 percent of internet connection in 2015 was from mobile phone [6].

FinTech can be roughly divided to 3 groups. First, innovative financial tool and service FinTech. This group tries to invent new financial tools and services by using benefit of the global internet. The global-level samples of well-known service providers in this group are Bitcoin [7], Paypal and

TransferWise [8]. Some of the services can eliminate roles of commercial bank from the whole business transactions. Second, algorithm related for executive support service (ESS) FinTech. This group usually analyses data by applying advanced algorithms, such as artificial intelligence and big data to get some valuable information from existing data which can point to new market targets. And, third, enhancement of traditional financial service FinTech. FinTech in this group can be considered as a support tool. The sample usages in this case are mobile stock trading via smart phone and online automatic credit application approval which will be discussed in this article.

As FinTech business has high value and relates to various stakeholders, both governments and international agencies have to apply law and regulation strictly on it. To avoid facing difficulties, many new FinTech tools have to be concerned about this legal issue before launching to real market [2].

More explanation of samples of FinTech and what should concern when using FinTech via the internet can be seen in the article that the authors wrote [3].

**Sample of FinTech supported by information technology and telecommunication**

This article will give an example of simple FinTech service that is the enhancement of traditional financial services in term of infrastructure usage and code of both client and server. The 2-tier software in this project separates jobs between client and server. Client is responsible for just user interface (UI) part, while server is responsible for business logic section as shown in Figure 1. Server is designed by using WCF services technique which is super set of standard web services (web services can work via just HTTP while WCF services can work via HTTP, TCP, MSMQ, P2P and named pipe). Result from the server can be used by clients those are created from several types of technology and platform.

When working via HTTP, WCF services has the same capability in firewall transparent as web services because it can use port number 80 (regular connection) or 443 (secured connection) as web services. Though WCF services can use other kinds of communication [9], such as TCP, P2P and named pipes, this article shows just HTTP due to the reason of web services compatibility and simplicity.

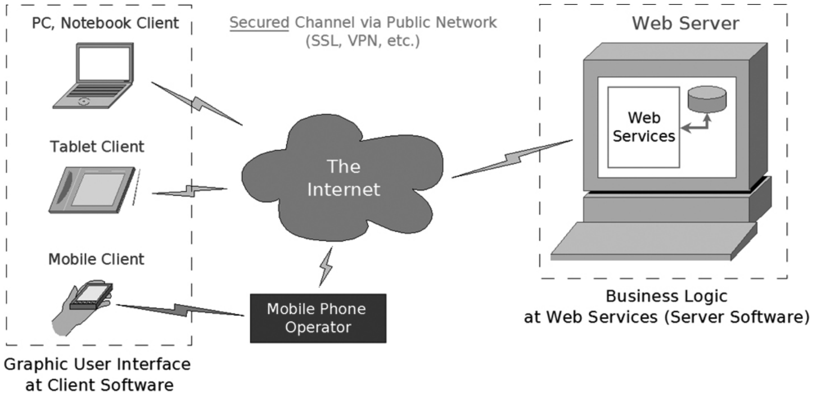


**Figure 1.** GUI is responsibility of clients whereas server monitoring the bussiness logic.

**FinTech for automatic screening consumer credit applications based on credit score**

In the past, consumer credit applications needed to be processed by the experienced staffs based on 5C's [10]. They had to consider the applicant's character, capital, collateral, capacity and condition. Though there were some efforts to quantify the characteristics, the credit approvals were rather subjective. Not just that, the cost of using expert for every transaction was also rather high.

With FinTech over the internet, consumer credit application form can be filled online. Credit scoring algorithms are coded as a program and this software is provided as a service (in this case, it is in WCF services platform). The client software just consumes the methods in this WCF services and the results of credit approval can be provided automatically. Credit scoring algorithms may be different from firm to firm, however, they are from the same concept and goal which is quantifying credit applicant's qualitative properties.

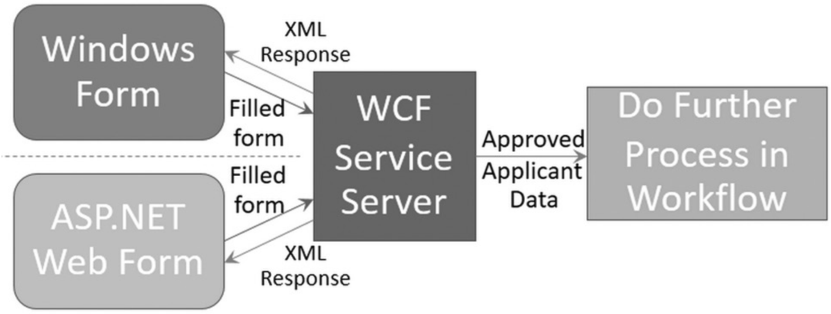


**Figure 2.** WCF service server gets data from multiplatform client and send the approved application data to next process in the company workflow.

This study creates basic sample FinTech software that can help the credit firm in screening credit application forms from information provided by customer. Credit application forms, in this project, can be in 2 platforms, classic windows form and ASP.NET web form. The windows form applications can be filled by credit company employees at the firm's office while the web form applications are filled by the credit applicants online or by credit company employees at the firm's office. As shown in Figure 2, clients in both platforms can send filled application form to the server that hosts WCF services. Business logic in WCF services will examine the submitted data. If the data pass the criteria, the applicant will be informed that the credit application is approved and his or her data will be sent to the next process in the company workflow, such as writing data in the database. The unqualified applicant will be notified that the application is rejected.

The credit rating formula used in the program comes from a classic book [11] published by The National Bureau of Economic Research. This sample calculation of credit score needs criteria from Table 1.

**Table 1.** Factors for calculate credit score [11]

| Factors | Score criteria |
|---|---|
| Age | Plus 0.01 for each year of age over 20, with the maximum of 0.30 for 50 years or more. |
| Gender | Plus 0.40 if applicant is a woman. |
| Stability of residence | Plus 0.042 for each year at present residence, with a maximum of 0.42 for 10 years or more. |
| Occupation | Plus 0.55 for either of two good-risk occupations (ex. teachers, nurses, doctors, technicians, lawyers, accountants), 0 for either of two bad-risk occupations (ex. outside salesman, wage earners) and 0.16 for all others. |
| Industry | Plus 0.21 for employee in utility industries, government service, bank and brokerage business. |
| Stability of employment | Plus 0.059 for each year of present employment, with a maximum of 0.59 for 10 years or more. |
| Bank account | Plus 0.45 |
| Real estate | Plus 0.35 |
| Life insurance | Plus 0.19 |

The sample 2-tier FinTech software system is going to demonstrate how to do basic online credit approval from credit scoring concept. Steps of working can be summarized as follows.

1. Credit application form, graphic user interface tier, can be filled in 2 platforms

○ Classic windows form platform: usually, this case is done at the office of credit company that has local area network (LAN). The firm's staff fills required data of the applicant in classic windows form application.

○ ASP.NET web form platform: using browser can be done either in the office of credit company or anywhere that can connect to the internet. The staff or the applicant fills required data in ASP.NET web form application via the browser.

2. The filled form in either platform sends data to WCF services server which is business logic tier.

3. WCF server that contains credit scoring algorithm scrutinizes the submitted factor data and responses to the client whether the credit application is accepted, rejected or still on hold to wait for decision of the expert. In this sample case, if the applicant gets score less than 1.5, the application will be rejected. If the score is greater than 2.5, the application will be accepted. If the score is between 1.5 and 2.5, the application will be held to wait for the expert to make a final decision as shown in Figure 3.

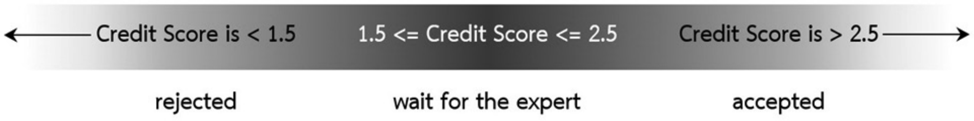


**Figure 3.** Criteria of accepted and rejected application based on sample credit score.

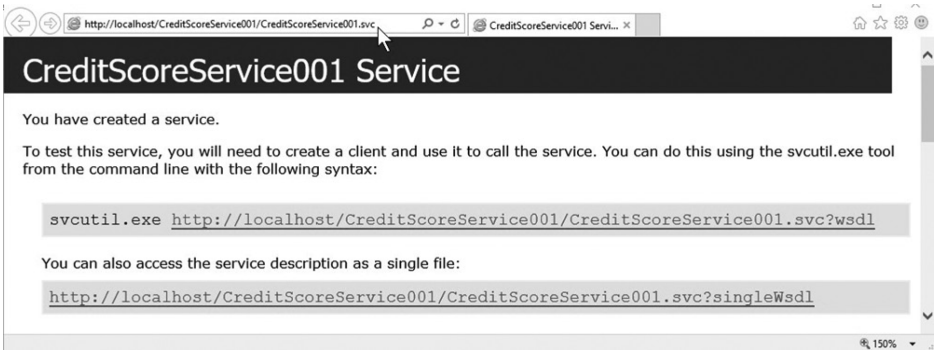
4.The accepted application's data are sent to the next process in the workflow, such as storing in the database.

**IT Technical related Issues**

Technical steps of this FinTech sample for enhancing traditional business via the internet can be separated to 2 sections. First, server section which can be called as WCF services server that may locate at the credit company. Second, client section that can be either windows or web application (or any types of technology that can consume WCF services).

For WCF services, ICreditScoreService001 interface ServiceContract contains 2 Operation Contract method signatures, CalculateCreditScore and CreditApprove. It's in ICreditScoreService001.cs file as shown in Listing 1. The interface's methods are implemented in CreditScoreService001.svc.cs as shown in Listing 2. CalculatCreditScore required the following data as its parameters, age, gender, year of present residence, year of present employment, occupation group, industry group, whether the applicant has bank account, real estate and life insurance. Occupation and industry group are used to differentiate risk level. Description and risk rate of the mentioned parameters are described in the classic book from NBER [12]. Parameter data are filled in the form in the format of TextBox, Radio Button and CheckBox controls.

Though WCF services can be described in many formats, this paper chooses Web Service Description Language (WSDL) as it conforms to the standard of web services. Figure 4 and 5 show samples of how to see and sample of WSDL gotten from this WCF services.


http://localhost/CreditScoreService001/CreditScoreService001.svc
CreditScoreService001 Servi... ×
150%

## CreditScoreService001 Service

You have created a service.

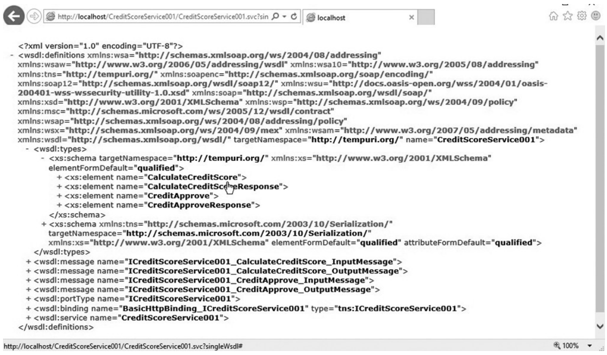
To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://localhost/CreditScoreService001/CreditScoreService001.svc?wsdl
```

You can also access the service description as a single file:

```
http://localhost/CreditScoreService001/CreditScoreService001.svc?singleWsdl
```

**Figure 4.** To get WSDL of WCF service, user needs to call the .svc file.



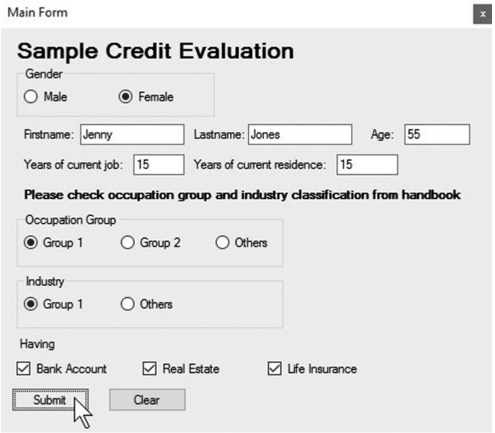
http://localhost/CreditScoreService001/CreditScoreService001.svc?sin
localhost

```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:wsa10="http://www.w3.org/2005/08/addressing"
  xmlns:tns="http://tempuri.org/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
  200401-wss-wssecurity-utility-1.0.xsd" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
  xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
  xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tempuri.org/" name="CreditScoreService001">
  - <wsdl:types>
    - <xs:schema targetNamespace="http://tempuri.org/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified">
        + <xs:element name="CalculateCreditScore">
        + <xs:element name="CalculateCreditScoreResponse">
        + <xs:element name="CreditApprove">
        + <xs:element name="CreditApproveResponse">
      </xs:schema>
    + <xs:schema xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/"
      targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/"
      xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="qualified">
    </wsdl:types>
  + <wsdl:message name="ICreditScoreService001_CalculateCreditScore_InputMessage">
  + <wsdl:message name="ICreditScoreService001_CalculateCreditScore_OutputMessage">
  + <wsdl:message name="ICreditScoreService001_CreditApprove_InputMessage">
  + <wsdl:message name="ICreditScoreService001_CreditApprove_OutputMessage">
  + <wsdl:portType name="ICreditScoreService001">
  + <wsdl:binding name="BasicHttpBinding_ICreditScoreService001" type="tns:ICreditScoreService001">
  + <wsdl:service name="CreditScoreService001">
</wsdl:definitions>
```

http://localhost/CreditScoreService001/CreditScoreService001.svc?singleWsdl#
100%

**Figure 5.** Sample of WSDL gotten from WCF services

This article shows 2 sample clients running in different platform, windows and web applications. Sample of the classic windows form client is shown in Figure 6. After the credit company staff fills the form and click submit, the result will be shown up in the format of message box as in Figure 7 and 8 is the sample ASP.NET web form application. All controls and concept are similar to the classic windows form version except the result is shown in label control, instead of message box.



Sample Credit Evaluation

Gender
○ Male ● Female

Firstname: Jenny Lastname: Jones Age: 55

Years of current job: 15 Years of current residence: 15

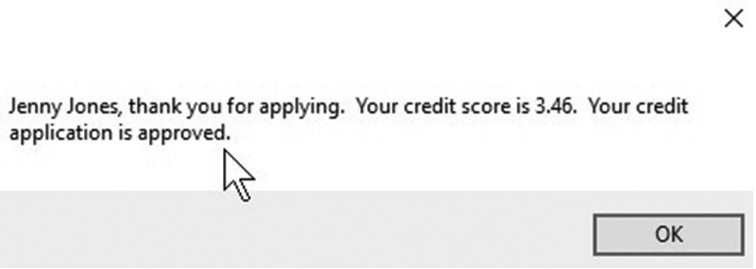
Please check occupation group and industry classification from handbook

Occupation Group
● Group 1 ○ Group 2 ○ Others

Industry
● Group 1 ○ Others

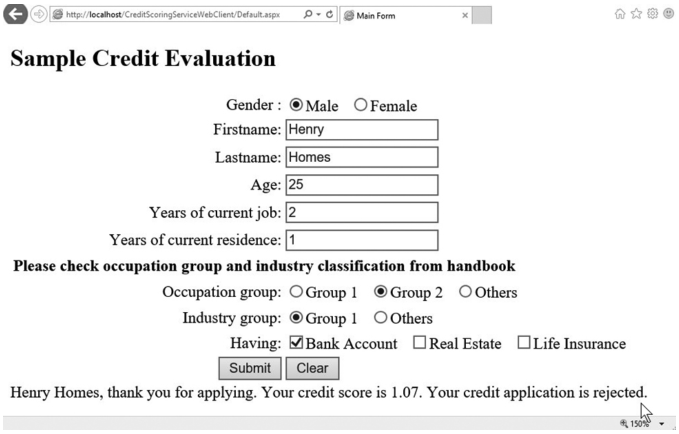
Having
☑ Bank Account ☑ Real Estate ☑ Life Insurance

**Figure 6.** Classic windows form client



Jenny Jones, thank you for applying. Your credit score is 3.46. Your credit application is approved.

**Figure 7.** Result in message box format



Sample Credit Evaluation

Gender : ● Male ○ Female
Firstname: Henry
Lastname: Homes
Age: 25
Years of current job: 2
Years of current residence: 1

Please check occupation group and industry classification from handbook

Occupation group: ○ Group 1 ● Group 2 ○ Others
Industry group: ● Group 1 ○ Others
Having: ☑ Bank Account □ Real Estate □ Life Insurance

Submit Clear

Henry Homes, thank you for applying. Your credit score is 1.07. Your credit application is rejected.

**Figure 8.** ASP.NET web form client which get result in label format

Partial codes of "CreditScoreService001" WCF services written in C# are listed in Listing 1.1 for the ServiceContract and Listing 1.2 for its interface's implementation. Listing 2 is the partial code of the web application client. Code of the windows application client is shown in Listing 3. All listing are located in the appendix. The complete codes of all 3 parts can be downloaded from "http://goo.gl/XaROuE". Both clients are also written in C# language as well.

## Conclusion

FinTech is the combination of financial services and technology. FinTech can be divided to 3 groups. First is enhancement of traditional financial services, such as online banking. Second is innovative financial tools and services, such as peer-to-peer lending. And, third is algorithm related for executive support services, such as big data analysis for credit card scoring. One reason that help FinTech grows up rapidly is the efficiency of network infrastructure, both computer and telecommunication network. This article also shows sample of basic 2-tier online credit application approval based on the data that user fills in. Using FinTech automatic processing like this can reduce cost of manual evaluation significantly. WCF services technology is chosen to operate all major business logic in the sample because it can be compatible with Web services open standard while there are some richer features. Clients, GUI part, can be built from types of technology that support WCF services and business logic part.

## References

1. Bryan B. The 10 hot new financial buzzwords you need to know. Business Insider [Internet]. 2015 [cited 2016 Jun 21]. Available from: http://www.businessinsider.com/the-10-hot-new-financial-buzzwords-you-need-to-know-2015-12
2. Laroche-Gray M. Growth in FinTech investment fastest in European market, according to accenture study. Accenture [Internet]. 2015 [cited 2016 Jun 8]. Available from: https://newsroom.accenture.com/industries/

financial-services/growth-in-fintech-investment-fastest-in-european-market-according-to-accenture-study.htm

3. Chantarasathaporn K, Ngammongkolwong S, Rattanaparinyanukool K. FinTech in enhancing traditional financial services with web services and multi-platform clients. Bus Rev 2016;8(2):199-218.
4. Einav L, Jenkins M, Levin J. The impact of credit scoring on consumer lending. RAND J Econ [Internet]. 2013 [cited 2016 Aug 15]. Available from: http://web.stanford.edu/~jdlevin/Papers/CreditScoring.pdf
5. Yu P. The next big deal at Wharton: FinTech. Wharton magazine [Internet]. 2015 [cited 2016 Aug 26]. Available from: https://beacon.wharton.upenn.edu/entrepreneurship/2015/07/the-next-big-deal-at-wharton-fintech/
6. Srivastava A. 2 billion smart phone users by 2015 : 83% of internet usage from mobiles [study]. Dazeinfo Inc. [Internet]. 2014 [cited 2016 Jun 8]. Available from: http://dazeinfo.com/2014/01/23/smartphone-users-growth-mobile-internet-2014-2017/
7. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. WwwBitcoinOrg [Internet]. 2008 [cited 2016 July 16]. Available from: https://bitcoin.org/bitcoin.pdf
8. TransferWise. How does TransferWise work and is it safe?. Telegraph Media Group Limited [Internet]. 2015 [cited 2016 Aug 13]. Available from: http://www.telegraph.co.uk/money/transferwise/how-does-it-work-and-is-it-safe/
9. Microsoft developer network. Comparing ASP.NET Web services to WCF based on development. [Internet]. 2016 [cited 2016 Sep 10]. Available from: https://msdn.microsoft.com/en-us/library/aa738737(v=vs.110).aspx
10. Gestel T , Baesens B. Credit risk management: basic concepts financial risk components, rating analysis, models, economic and regulatory capital. Oxford: Oxford University Press; 2009.
11. Durand D. Credit-rating formulae. In: risk elements in consumer instalment financing. The National Bureau of Economic Research [Internet]. 1941 [cited 2016 Sep 15]. Available from: http://www.nber.org/chapters/c9265
12. Durand D. Findings of risk factor studies. In: risk elements in consumer instalment financing. The National Bureau of Economic Research [Internet]. 1941 [cited 2016 Sep 15]. Available from: http://www.nber.org/chapters/c9264

## Appendix

**Listing 1.1** "ICreditScoreService001" ServiceContract in BasicSampleCarInsuranceService.cs (WCF Services Server Part)

```
  using System.ServiceModel;
  namespace CreditScoreService001Prj  {
          [ServiceContract]
          public interface ICreditScoreService001  {
               [OperationContract]
               double CalculateCreditScore(int intAge, bool blnFemale, int intYearPresentResidence, int
intOccupationGroup, int intIndustry, int intYearPresentEmployment, bool blnBankAccount, bool blnRealEs
tate, bool blnLifeInsurance);
          [OperationContract]
          string CreditApprove(double dblCreditScore);
      }
  }
```

**Listing 1.2** "CreditScoreService001" class implements ICreditScoreService001 interface in CreditScore Service001.svc.cs (WCF Services Server Part)

```
namespace CreditScoreService001Prj {
    public class CreditScoreService001 : ICreditScoreService001 {
        static string strFn, strLn;
       public double CalculateCreditScore(string strFirstname, string strLastname, int intAge, bool blnFemale, int intYear-
PresentResidence, int intOccupationGroup, int intIndustry, int intYearPresentEmployment, bool blnBankAccount, bool
blnRealEstate, bool blnLifeInsurance) {
            strFn = strFirstname;  strLn = strLastname;
            double dblScore = 0;
            // Plus 0.01 for each year of age over 20, with the maximum of 0.30 for 50 years or more.
            dblScore += ((intAge >= 50) ? 0.3 : (0.01 * intAge));
            // Plus 0.40 if applicant is a woman.
            dblScore += ((blnFemale == true) ? 0.40 : 0);
            // Plus 0.042 for each year at present residence, with a maximum of 0.42 for 10 years or more.
            dblScore += ((intYearPresentResidence >= 10) ? 0.42 : (0.042 * intYearPresentResidence));
           // Plus 0.55 for either of two good-risk occupations (ex. teachers, nurses, doctors, technicians, lawyers, accoun-
tants), 0 for either of two bad-risk occupations (ex. Outside salesman, Wage earners) and 0.16 for all others.
            dblScore += ((intOccupationGroup == 1) ? 0.55 : ((intOccupationGroup == 2) ? 0 : 0.16));
            // Plus 0.21 for employee in utility industries, government service, bank and brokerage business.
            dblScore += ((intIndustry == 1) ? 0.21 : 0);
            // Plus 0.059 for each year of present employment, with a maximum of 0.59 for 10 years or more.
            dblScore += ((intYearPresentEmployment >= 10) ? 0.59 : (0.059 * intYearPresentEmployment));
            // Plus 0.45 if having bank account
            dblScore += (blnBankAccount ? 0.45 : 0);
            // Plus 0.35 if having real estate
            dblScore += (blnRealEstate ? 0.35 : 0);
            // Plus 0.19 if having life insurance
            dblScore += (blnLifeInsurance ? 0.19 : 0);
            return dblScore;
        }
        public string CreditApprove(double dblCreditScore) {
            string strCreditApprove = "";
            strCreditApprove = ((dblCreditScore < 1.5) ? "rejected" : ((dblCreditScore > 2.5) ? "approved" : "submitted to the expert and you will be contacted back by 16.00 tomorrow"));
            NotifyApprovedApplication(strFn, strLn);
            return strCreditApprove;
        }
        private static void NotifyApprovedApplication(string strFn, string strLn) {
            // do further process in company workflow for approved application.
        }
    }
}
```

Listing 2. WCF services client in ASP.NET web form C# code

```
using System;
using System.Web.UI;
using CreditScoreServiceProxy;
public partial class _Default : Page {
    bool blnFemale, blnBankAccount, blnRealEstate, blnLifeInsurance;
    string strFirstname, strLastname;
    int intAge, intYearOfCurrentJob, intYearOfCurrentResidence, intOccupationGroup, intIndustryGroup;
    double dblCreditScore = 0;
    string strCreditApprove = "";
    protected void Page_Load(object sender, EventArgs e) {
        if (!Page.IsPostBack) {
            strFirstname = strLastname = "";
            blnFemale = blnBankAccount = blnRealEstate = blnLifeInsurance = false;
            intAge = intYearOfCurrentJob = intYearOfCurrentResidence = intOccupationGroup = intIndustryGroup = 0;
        }
    }
    protected void btnClear_Click(object sender, EventArgs e) {
        rdoMale.Checked = false;
        rdoFemale.Checked = true;
        txtFirstname.Text = txtLastname.Text = "";
        txtAge.Text = txtYearOfCurrentResidence.Text = txtYearOfCurrentJob.Text = "0";
        rdoOccupationGroup1.Checked = rdoOccupationGroup2.Checked = false;
        rdoOccupationGroupOthers.Checked = true;
        rdoIndustryGroup1.Checked = false;
        rdoIndustryGroupOthers.Checked = true;
        chkBankAccount.Checked = chkRealEstate.Checked = chkLifeInsurance.Checked = false;
        lblResult.Text = "";
    }
    protected void btnSubmit_Click(object sender, EventArgs e) {
        CreditScoreService001Client serviceClient = new CreditScoreService001Client();
        blnFemale = ((rdoFemale.Checked) ? true : false);
        strFirstname = txtFirstname.Text;
        strLastname = txtLastname.Text;
        intAge = int.Parse(txtAge.Text);
        intOccupationGroup = ((rdoOccupationGroup1.Checked) ? 1 : ((rdoOccupationGroup2.Checked) ? 2 : ((rdoOccupationGroupOthers.Checked) ? 3 : 0)));
        intIndustryGroup = ((rdoIndustryGroup1.Checked) ? 1 : ((rdoIndustryGroupOthers.Checked) ? 2 : 0));
        intYearOfCurrentJob = int.Parse(txtYearOfCurrentJob.Text);
        intYearOfCurrentResidence = int.Parse(txtYearOfCurrentResidence.Text);
        blnBankAccount = ((chkBankAccount.Checked) ? true : false);
        blnLifeInsurance = ((chkLifeInsurance.Checked) ? true : false);
        blnRealEstate = ((chkRealEstate.Checked) ? true : false);
        dblCreditScore = serviceClient.CalculateCreditScore(intAge, blnFemale, intYearOfCurrentResidence, intOccupationGroup, intIndustryGroup, intYearOfCurrentJob, blnBankAccount, blnRealEstate, blnLifeInsurance);
        strCreditApprove = serviceClient.CreditApprove(dblCreditScore);
        serviceClient.Close();
        lblResult.Text = string.Format("{0} {1}, thank you for applying.  Your Credit Score is {2}.  Your credit application is {3}.", strFirstname, strLastname, dblCreditScore.ToString(), strCreditApprove);
    }
}
```

**Listing 3.** WCF services client in windows form C# code

```
using System;
using System.Windows.Forms;
using CreditScoringServiceWinClientPrj.CalculateCreditScoreServiceRef;
namespace CreditScoringServiceWinClientPrj {
    public partial class MainForm : Form {
        bool blnFemale, blnBankAccount, blnRealEstate, blnLifeInsurance;
        string strFirstname, strLastname;
        int intAge, intYearOfCurrentJob, intYearOfCurrentResidence, intOccupationGroup, intIndustryGroup;
        double dblCreditScore = 0;
        string strCreditApprove = "";
        CreditScoreService001Client serviceClient = null;
    public MainForm() {
        InitializeComponent();
        strFirstname = strLastname = "";
        blnFemale = blnBankAccount = blnRealEstate = blnLifeInsurance = false;
        intAge = intYearOfCurrentJob = intYearOfCurrentResidence = intOccupationGroup = intIndustryGroup = 0;
    }
    private void btnClear_Click(object sender, EventArgs e) {
        rdoMale.Checked = false;
        rdoFemale.Checked = true;
        txtFirstname.Text = txtLastname.Text = "";
        txtAge.Text = txtYearOfCurrentResidence.Text = txtYearOfCurrentJob.Text = "0";
        rdoOccupationGroup1.Checked = rdoOccupationGroup2.Checked = false;
        rdoOccupationGroupOthers.Checked = true;
        rdoIndustryGroup1.Checked = false;
        rdoIndustryGroupOthers.Checked = true;
        chkBankAccount.Checked = chkRealEstate.Checked = chkLifeInsurance.Checked = false;
    }
    private void btnSubmit_Click(object sender, EventArgs e) {
        serviceClient = new CreditScoreService001Client();
        blnFemale = ((rdoFemale.Checked) ? true : false);
        strFirstname = txtFirstname.Text;
        strLastname = txtLastname.Text;
        intAge = int.Parse(txtAge.Text);
        intOccupationGroup = ((rdoOccupationGroup1.Checked) ? 1 : ((rdoOccupationGroup2.Checked) ? 2 : ((rdoOccupationGroupOthers.Checked) ? 3 : 0)));
        intIndustryGroup = ((rdoIndustryGroup1.Checked) ? 1 : ((rdoIndustryGroupOthers.Checked) ? 2 : 0));
        intYearOfCurrentJob = int.Parse(txtYearOfCurrentJob.Text);
        intYearOfCurrentResidence = int.Parse(txtYearOfCurrentResidence.Text);
        blnBankAccount = ((chkBankAccount.Checked) ? true : false);
        blnLifeInsurance = ((chkLifeInsurance.Checked) ? true : false);
        blnRealEstate = ((chkRealEstate.Checked) ? true : false);
        dblCreditScore = serviceClient.CalculateCreditScore(intAge, blnFemale, intYearOfCurrentResidence, intOccupationGroup, intIndustryGroup, intYearOfCurrentJob, blnBankAccount, blnRealEstate,
blnLifeInsurance);
            strCreditApprove = serviceClient.CreditApprove(dblCreditScore);
            serviceClient.Close();
            MessageBox.Show(string.Format("{0} {1}, thank you for applying.  Your Credit Score is {2}.
Your credit application is {3}.", strFirstname, strLastname, dblCreditScore.ToString(), strCreditAp-
prove));
        }
    }
}
```