

Mohamad Jawad

REAL-ESTATE INTERFACE

REAL-ESTATE INTERFACE

Mohamad Jawad
Bachelor's thesis
Spring 2016
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Software Engineering

Author: Mohamad Jawad

Title of the bachelor's thesis: Real – estate Interface

Supervisor: Risto Korva

Term and year of completion: Spring 2016

Number of pages: 28

The purpose of the thesis was to implement the most efficient user interface (UI) for Real-estate in Finland for client companies due to their desire of having this feature in their system. The prototype was supposed to show the clients how the feature works to get needed data for real-estate properties in Finland in their map system.

National Land Survey MML of Finland was chosen for tracking the real-estate properties data in the system. The real-estate prototype was developed by Microsoft Development tools. The client sends a request for certain data to the cloud that resends the request to the MML service through Application map software which had been developed earlier.

As a result, the prototype is properly implemented into Client's map software. After this thesis the development of the prototype has been continued

Keywords: Interface, HTTPS, MML, client's cloud (OVT), Map, Data, Real-estate.

PREFACE

This thesis was commissioned for the client company and done during 1 February – 28 April 2016. Clients ordered the Real-estate interface product. The role of the instructor was guiding the thesis from the requirements and bases of writing a thesis document through meetings. The role of the supervisor was instructing the thesis plan and its requirements which were done by the author in addition to testing the Real – estate interface feature in the client’s application.

I would like to thank my Thesis’s instructor Risto Korva and the clients for instructing and guiding my thesis and their great influence and support in my thesis which has resulted to a success.

Oulu, 20.4.2016
Mohamad Jawad

CONTENTS

ABSTRACT	3
PREFACE	4
TABLE OF CONTENTS	5
VOCABULARY	6
1 INTRODUCTION	7
2 OPERATIONAL ENVIRONMENT	8
2.1 Sawmill	8
2.2 National Land Survey of Finland (MML)	9
2.3 Client's application	10
2.4 System outline	11
3 MML - OVT INTERFACE	13
3.1 Map coordinates and registry number	13
3.2 HTTP Request / Response	15
3.3 XML	16
3.4 Server Security	19
4 APPLICATION INTERFACE	23
4.1 Client's interface	23
4.2 Invoicing	25
5 SUMMARY	27
REFERENCES	28

VOCABULARY

- **Cadastre:** Is a comprehensive register of the real estate.
- **ERP:** Enterprise Resource Planning.
- **HTML:** Hyper Text Markup Language.
- **HTTP:** Hyper Text Transfer Protocol.
- **HTTPS:** Hyper Text Transfer Protocol for secure.
- **MML:** National Land Survey of Finland (Maanmittauslaitos).
- **MapWinGIS:** MapWinGIS is a function in Visual Studio which is used to provide a GIS and mapping functionality to any Windows Forms based application.
- **OVT:** Client's Cloud.
- **RE:** Real – estate.
- **SQL:** Structured Query Language.
- **SSL:** Secure Socket Layer.
- **UI:** User Interface
- **URL:** Uniform Resource Locator.
- **VB:** Visual Basic.
- **XML:** Extensible Markup Language.
- **XSD:** Schema Definition Language.

1 INTRODUCTION

The purpose of this study was to find out the most efficient and effective solution for the Real-estate Service Procedure in the sawmill Industry. It should provide the client user general information, dimensions and invoicing for a certain property.

Clients are the companies which specialize in wood purchases and processing system such as contracts, wood modifications (cutting, modifying), transportations, logistics, etc. In addition, the system provides tools for planning and settings and monitoring.

Clients have ordered real-estate interface feature due to their desire of having this feature in their system. The real – estate features allows them to have a general data information about properties in Finland through the map system, for example the owner name of the property.

The topic of this thesis was implemented in the client's application and based on the maps feature which already existed and had been developed earlier.

2 OPERATIONAL ENVIRONMENT

2.1 Sawmill

The purpose of the Real-estate property Service is to have right information about a certain property for business implementation from buying the property and manufacturing the wood.

The sawmilling business varies according to their operating strategy. The basic idea is the same. The process starts with sawmill logs measurements from the length of the log, diameter, volume and the quality of the crookedness. Based on these measurements data the log is paid to the dealer and shipping organization. This step allows having a variety of sawing patterns of desired optimal measurements that achieves the best value of the yield. Then the logs are sorted into appropriate classes ready for cutting.

The sawing procedure varies according to economic factors. It is possible to choose different ways of cutting the logs, for example a circular sawing blade process where the blades can be on the same axis and the number of the device unit that may allow the overlapping of the log. Circular sawing blade diameter determines the maximum height of the sawn log.

When the logs are sawn, they are sorted according to their dimensions from their width and thickness. This step allows drying them in loads since it provides a good air circulation in different parts of the load. When the drying has been finished, it is moved to a forklift kiln drying according to customer wishes. It consists of two types, continuous drying and batch kilns. In continuous drying the boards are still moist and they are fed to the upstream end of the channel along the strap. The channel can be up to a 10 - 15 ° C temperature (1). In kiln drying the boards are places in kiln with temperature about 70 ° C (1). Conditions vary according to the desired result. After that, they are sorted into dimensions and quality classes. The classification is defined according to the moisture percentage and the length of the boards (1).

2.2 National Land Survey of Finland (MML)

The National Land Survey of Finland is a service that maintains an official purchase price register based on real estate transactions. The aim is to be of service in defining property values for expropriations and land use planning and for providing credit, and in other evaluation and investigation tasks. The records of the purchase price register carry a fee determined by the price list of the National Land Survey (4). The cadastre is a public basic register, which is a part of the Land Information System in MML. It contains real property information. The cadastre contains the following information about real property units and other register units:

- Real property unit identification number (property identifier).
- Area.
- Shares in joint property units (e.g. water areas).
- Information about rights of usufruct and restrictions on use that apply to real property units (4).

The cadastral registry location data is presented on the cadastral index map. The boundaries of real property units and other register units, as well as identifying information (the real property unit identification number or a part of it) are shown on the cadastral index map.

Privacy is a very important role in using the MML service system, since it contains private data of the public. Thus, it is confidential to keep this privacy safe from the public because personal data of real-estate properties is fetched. In this topic the MML service system is used to receive the owner's name of a certain real-estate property.

MML offers certain SQL queries to fetch data information of land properties. Developers do not have the right to update or fetch all the data of a certain property unless they have a license for that. In this implementation MML offered SQL queries to fetch the needed data of the land properties, and they are then presented in the client's application.

2.3 Client's application

The client's applications is an electronic applications that applies trading and real-life harvesting and transportations procedure, which allows them to have a control of managing the wood business online.

The MML interface service allows obtaining up-to-date data directly from the information providers. Thus, the client's application does not have to acquire the data sets and deal about storing and updating the data. The interface service provides the application with: Property information, Topographic maps and aerial photographs, Nomenclatures and Building information through SQL queries which the customer run the queries in order to get the information of the real-estate property.

The MML service is being implemented in the customer application interface so that the user is able to click on a certain area or land property in order to get the information needed. Also, an invoicing log is created in the client's system to check or count how many requests the user has done for fetching data. Based on that, payment bills are calculated and payed directly through the system invoicing tool. The right of saving the records (for example owner names) in the customer's system is prohibited by a confidential law in Finland regarding the personal data of the public.

2.4 System outline

The key of a successful project is planning. Creating a project plan is the first thing to do when undertaking any project (5), by getting to know all aspects that are considered during the implementation of the project and having the general expectation of the project results. The main aspects that are taken in the project are:

- 1- Project goals.
- 2- Project deliverables.
- 3- Technical schema.

(Figure 1.)

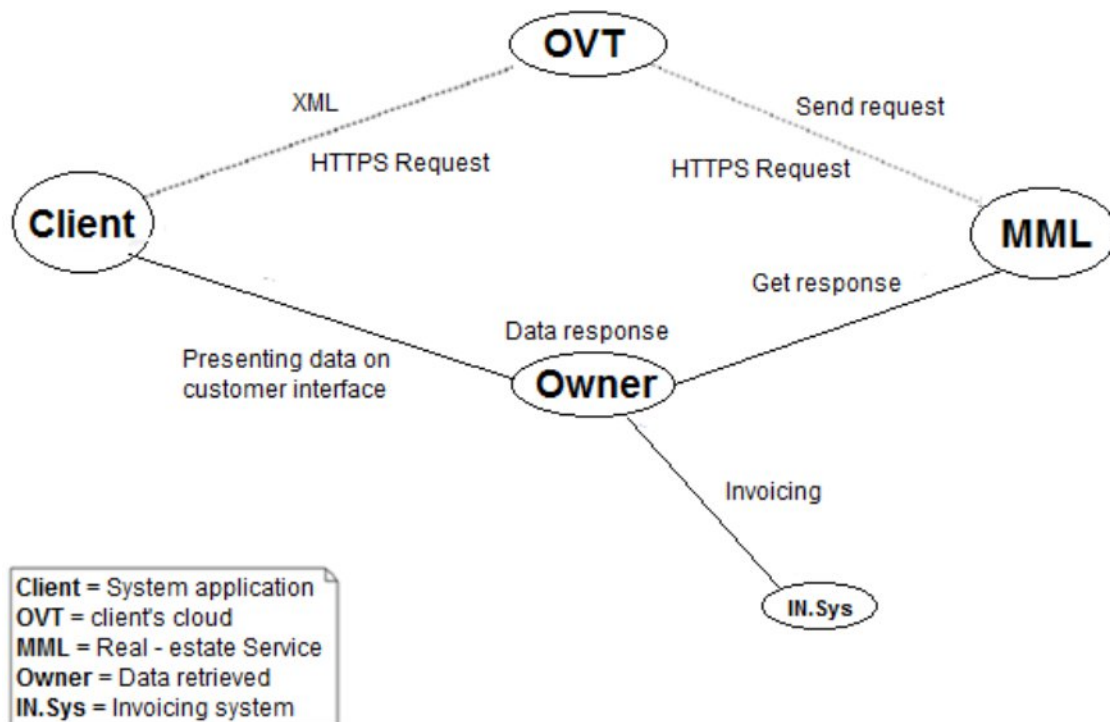


FIGURE 1. Plan diagram for a real-estate interface project

A request is sent by the client to the cloud after a click function has been triggered on certain coordination of the interface map. After OVT has received the request from the client, a registry number is fetched of that area through the coordinates that are clicked. Every certain area coordinates of Finland are hold by a registry number which is given by the Finnish government. Through the registry number of the area the request is sent from OVT to the MML service URL where the response is retrieved as a stream holding the owner name and basic information of the area and it is presented in the client's interface. Security is not neglected when sending requests through the web-services using https requests instead of http to avoid any type of attack and thus protecting the information that is transferred from MML to the client's server. User authentication is also considered so that a username and a password are required in order to send a single request to the customer Cloud.

Invoicing and logging are implemented as a part of the thesis since invoicing criteria is based on the total number of requests that the client sends to the OVT server. A log database is created to save every request that the client has sent. User identification is the private key used for saving where the request type is followed so that every client is presented by their own identification. Saving the data, which has been fetched as a response from MML to the OVT server, has not been implemented due to the Finnish law restrictions.

In a nutshell, the process of system was implemented as:

- The user clicks on the area coordinates in the map.
- The user enters the authentication (a username and a password).
- MML sends the request to OVT.
- OVT re-sends automatically the request to MML through the area registry number.
- MML gives back the response.
- OVT gets the response.
- OVT sends the response as a file format.
- OVT saves the request into the database for Invoicing.

3 MML - OVT INTERFACE

3.1 Map coordinates and registry number

Using the .NET Language a basic method is created for a real-estate for fetching data that holds all the steps of handling the requests between OVT and MML. The coordinates are fetched according to the area that is selected by the client through the PixelsToProj function which .NET Visual studio has provided for the developers. Basic variables are created as pixels and projections and defined as integers and decimals (Formula 1.).

Public Property PixelX As Int64

Public Property PixelY As Int64

Public Property ProjX As Decimal

Public Property ProjY As Decimal

FORMULA 1

MML data is provided on certain coordinates on the map through sheets that are installed into the map through the MapWinGIS function that .NET has provided. MML sheets are formatted as DBF which stands for a database file, and they are presented on the map as shapes. Each shape has its own registry number that is needed to fetch the data through the MML webserver. The coordinates which MML has provided, are very small so zooming is needed in order to make them clearly visible to human eyes. In this case, a projection is drawn a distance of 13m in the scale of 1/15000 from the MML coordinates to vertical direction. Then bounds are set through the point to form an imaginary rectangular shape, which is not visible to (Figure 2.).

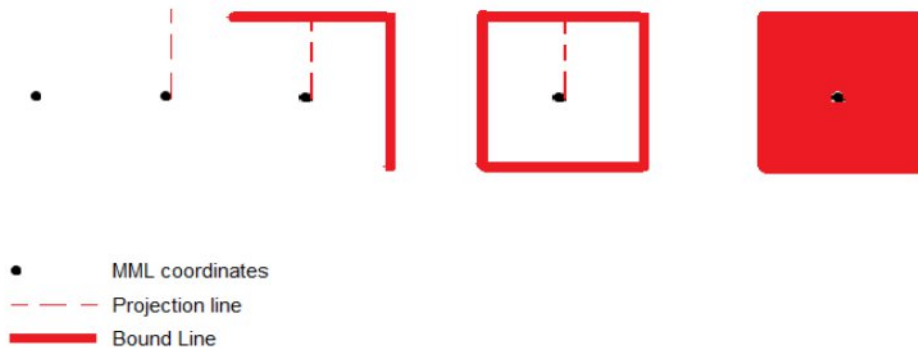


FIGURE 2. MML imaginary shapes formation as buttons inputs.

The aim of drawing an imaginary shape around the MML coordinates was to ease the usability of the user. When certain coordinates are clicked near the MML coordinates, a looping function is generated through the all coordinates inside the imaginary shape in order to retrieve the register number of the MML coordinates. The register number is formatted as: XXX-XXX-X-XX (Figure 3.)

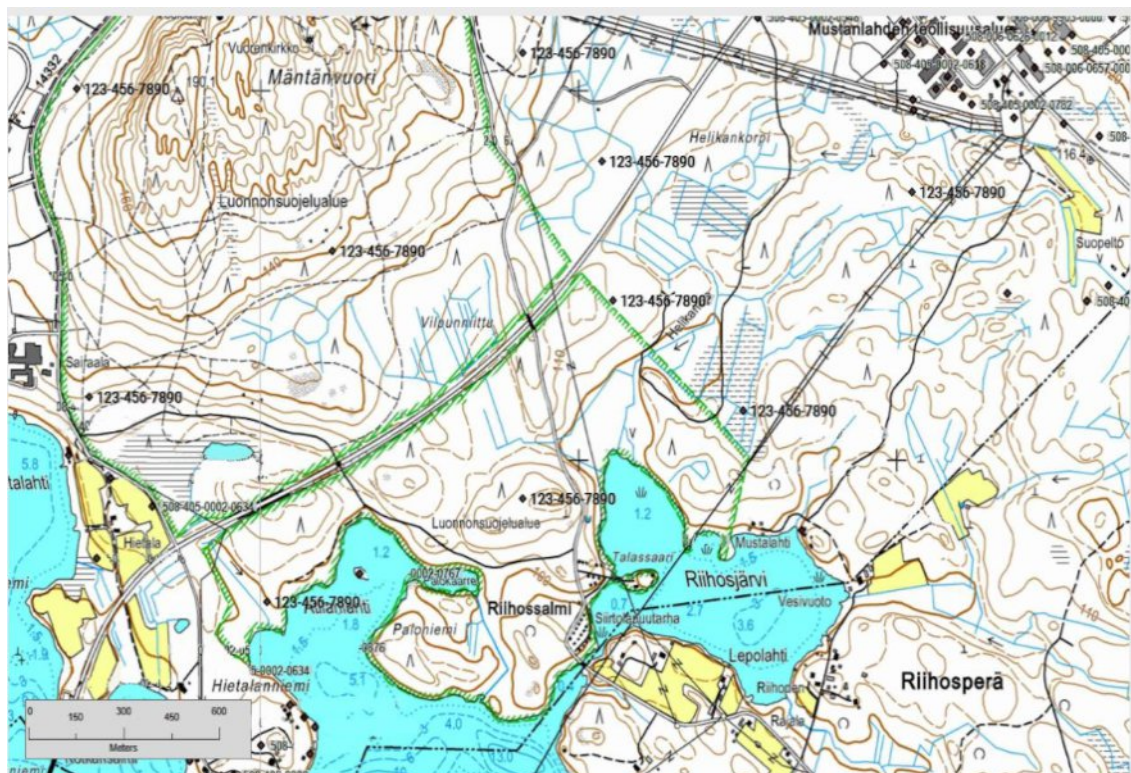


FIGURE 3. Coordinates installation on the client's application.

3.2 HTTP Request / Response

HTTP is a request/response protocol, which means that the client's computer sends a request for some file (e.g. "Get me the file 'home.html'"), and the web server answers back a response ("Here's the file", followed by the file itself) (6). A basic web request is created to the remote site as an instance using the system .Net function and then giving the request parameter of the MML URL server and the register number, which comes at the end of the URL as a parameter (Formula 2.).

```
Dim Request As System.Net.HttpWebRequest (3)
```

```
Request= System.Net.HttpWebRequest.Create
```

```
("https://www.mml.fi/123-123-3-11")
```

FORMULA 2

In order to get the response back, the remote site is called to retrieve the data in a response as an object (Formula 3.).

```
Dim response As System.Net.HttpWebResponse = request.GetResponse() (3)
```

FORMULA 3

The .StatusCode function is used to check if data is fetched correctly without errors so that the register number is determined if it is formatted correctly or not. There are different ways to parse the response into data for a user interface. In this case, the response is parsed to stream data and put it into a file under the name of the current date and time. Data is transmitted between servers through the HTTP protocol (Figure 4.)

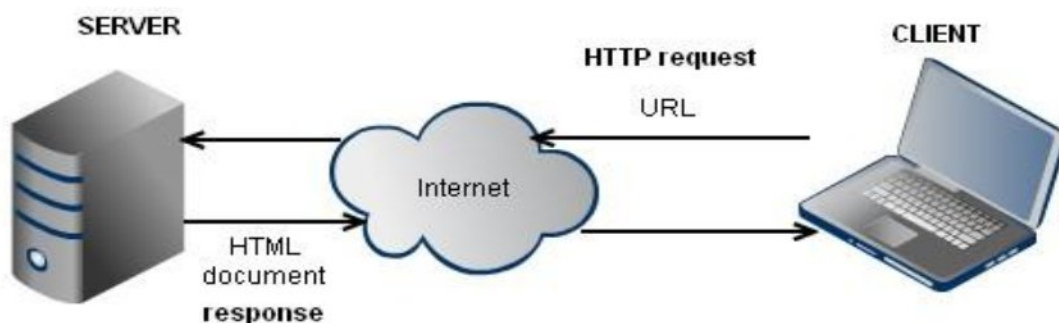


FIGURE 4. Data transmission between clouds through the HTTP protocol (1)

3.3 XML

XML is a self-describing language and it gives data as well as rules to identify what information it contains. Like HTML, XML is a subset of the SGML - Standard Generalized Markup Language (11). The following links give you more information about XML Files and its operations through VB.NET.

XML is a general purpose tag based language and it is very easy to transfer and store data across applications. XML files are made up of tags, which contain data, generally a start tag and an end tag to hold the data. For example, if you want to create an XML tag named "Header", the start tag is like **< Header >** and the end tag is like **</Header >**, where information can be filled between these tags. XML is case sensitive and tags must be closed in the reverse order that they were opened (11).

In this case, data is provided by certain queries that MML has offered as an XML file, which holds the real-estate owner's data. A basic XML document is presented as code (Formula 4.).

```
<kypt:Perustiedot xmlns:y="http://xml.nls.fi/ktjkir/yhteinen/2014/10/01" rekisteri-tilannepvm="2015-03-18">
```

```
  <kypt:Laitos>
```

```
    <trpt:laitoksenPerustiedot>
```

```
      <y:laitostunnus>85841600030101L0001</y:laitostunnus>
```

```
      <trpt:erityisenOikeudenLaji>0604</trpt:erityisenOikeudenLaji>
```

```
        <trpt:olotila>1</trpt:olotila>
```

```
        <trpt:voimassaolonPaattymispvm>2026-01-
```

```
        19</trpt:voimassaolonPaattymispvm>
```

```
        <trpt:voimassaololaji>01</trpt:voimassaololaji>
```

```
        <trpt:eoHenkilot>
```

```
          <y:Henkilo henkilorooli="EOVA" lukuuntoiminnanlaji="OL">
```

```
            <y:henkilonTiedot>
```

```
              <y:henkilolaji>LU</y:henkilolaji>
```

```
              <y:henkilotunnus>xxx</y:henkilotunnus>
```

```
              <y:etunimet>xxx</y:etunimet>
```



```

        <y:syntymapvm>xxx</y:syntymapvm>
        <y:ulkomaalainen>>false</y:ulkomaalainen>
        </y:henkilonTiedot>
        </y:Henkilo>
        </trpt:eoHenkilot>
    </trpt:laitoksenPerustiedot>
</kypt:Laitos>
</kypt:Perustiedot> (4)

```

FORMULA 4

The XML serialization is the process of converting an object into a form that can be readily transported. It serializes only public fields and property values of an object into an XML stream (11). .NET provides an XMLSERIALIZER class which creates vb.net files and compiles them into .dll files to perform a serialization. The XML serialize class contains all serialize/ de-serialize methods. It also generates the XML stream, which is compliant with World Wide Web consortium XML schema definition XSD. A deserialization is the reverse process of a serialization in which an object is retrieved from a disk based format (Formula 5.). It reads an XML document and constructs an object that is typed to the XML schema XSD of the document. The XML serialize class must be always constructed using a type of object in addition to a stream reader.

```
Private Sub DeserializeObject(filename As String)
```

```
    Console.WriteLine("Reading with TextReader")
```

```
    ' Create an instance of the Xml serialize object of specifying type.
```

```
    Dim serializer As New XmlSerializer(GetType(OrderedItem))
```

```
    ' Create a Text reader to read the file.
```

```
    Dim fs as New FileStream(filename, FileMode.OpenOrCreate)
```

```
    Dim reader As New StreamReader(fs)
```

```
    ' Declare an object variable of the type to be de-serialized.
```

```
    Dim i As OrderedItem
```

' Use the Deserialize method to restore the object's state.

```
i = CType(serializer.Deserialize(reader), OrderItem)
```

' Write out the properties of the object.

```
Console.WriteLine(i.ItemName & ControlChars.Tab & _  
    i.Description & ControlChars.Tab & _  
    i.UnitPrice & ControlChars.Tab & _  
    i.Quantity & ControlChars.Tab & _  
    i.LineTotal) (3)
```

End Sub

FORMULA 5

In MML the response is streamed and then passed to a stream reader in order to get an XML document as a string and then the XML serialize object is being able to de-serialize the document into an object class noting that the class hierarchy must have the same format of the XML response. Many online browsers provide the generator of constructing an object class through an XML document where the XML document is converted into the XSD form and then XSD can be converted into a class (Formula 6).

Partial Public Class Henkilo

```
Private henkilonTiedotField As HenkiloHenkilonTiedot
```

```
Public Property henkilonTiedot() As HenkiloHenkilonTiedot
```

```
Get
```

```
Return Me.henkilonTiedotField
```

```
End Get
```

```
Set
```

```
Me.henkilonTiedotField = value
```

```
End Set
```

```
End Property
```

End Class

Partial Public Class HenkiloHenkilonTiedot

```

Private etunimetField As String
Private sukunimiField As String
Public Property etunimet() As String
    Get
        Return Me.etunimetField
    End Get
    Set
        Me.etunimetField = value
    End Set
End Property
Public Property sukunimi() As String
    Get
        Return Me.sukunimiField
    End Get
    Set
        Me.sukunimiField = value
    End Set
End Property
End Class (4)

```

FORMULA 6

3.4 Server Security

A secure server is a web server, which guarantees secure online transactions. Secure servers use the Secure Sockets Layer (SSL) protocol for data encryption and decryption to protect data from unauthorized interception (2). Servers provide a variety of internal and external user services in organizations, which are constantly vulnerable to security threats, due to data sensitivity. Secure servers help organizations and businesses conduct secure and private network transactions. Until recently, e-commerce opportunities were often lost because of online user security concerns. However, the growth of online retailing has expanded requirements for security and measures geared toward preventing malicious attacks.

Hacking is an unauthorized intrusion into a computer or a network. The person engaged in hacking activities is generally referred to as a hacker. This hacker may alter system or security features to accomplish a goal that differs from the original purpose (7).

Hackers employ a variety of techniques for hacking, including:

- **Vulnerability scanner:** checks computers on networks for known weaknesses.
- **Password cracking:** the process of recovering passwords from data stored or transmitted by computer systems.
- **Viruses:** self-replicating programs that spread by inserting copies of the same program into other executable code files or documents.

Microsoft Windows Communication Foundation (WCF) provides these security features by default for any application that is built on top of the WCF framework.

A key security feature includes:

- **Auditing.** Effective auditing and logging is the key to non-repudiation. Non-repudiation guarantees that a user cannot deny performing an operation or initiating a transaction (3).
- **Authentication.** Authentication allows you to confidently identify the clients of your service. These might be end users, other services, processes, or computers. WCF supports mutual authentication, which identifies both the client and the service in tandem, to help in preventing man-in-the-middle attacks.
- **Authorization.** Authorization determines what system resources and operations can be accessed by the authenticated user. This allows you to grant specific application and resource permissions for authenticated users.
- **Confidentiality.** Confidentiality, also referred to as privacy, is the process of making sure that data remains private and confidential, and that it cannot be viewed by unauthorized users. Encryption is frequently used to

enforce confidentiality. Privacy is a key concern, particularly for data/messages passed across networks (3).

- **Integrity.** Integrity is the guarantee that data is protected from accidental or deliberate modification. Like privacy, integrity is a key concern, particularly for data/messages passed across networks. Integrity for data in transit is typically provided by using hashing techniques and message authentication codes (3).

Based on the MML Interface, security is considered due to privacy of the data such as sending requests over HTTPS and user authentication. HTTPS is significantly more secure than HTTP. The main three differences are that HTTP does not encrypt. HTTPS encrypts all the data going to the server (Figure 5.). In either case the attacker can intercept the data between you and the server, but the encryption with HTTPS ensures that attackers cannot do anything with it. HTTP does not care what server you connect to. HTTPS verifies the server against the certificate. If the user makes a double click on the padlock icon in Internet Explorer, they can see who the server belongs to and they can be confident that card details are going to the right people. Better than that if the server address and certificate disagree, the browser will give a warning that attackers do not match and this gives you the opportunity to leave before giving any information away (6). HTTP pages are stored on your computer and Internet caches. The pages load faster, but they are stored on systems that the user potentially does not have control over (i.e. your ISP's caching proxy or a computer in an Internet cafe). HTTPS pages are not cached anywhere. Therefore, there are no copies of your data hanging around to be retrieved later. Figure 6 shows the difference between HTTP and HTTPS.

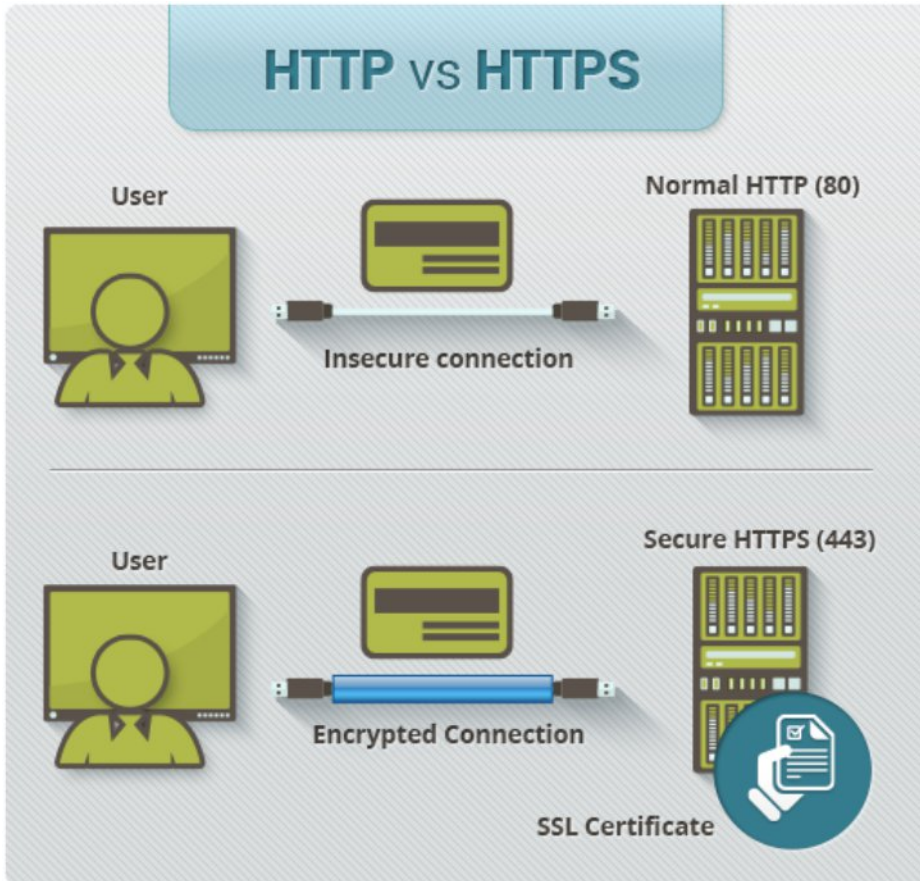


FIGURE 5. Data transmission security between HTTP and HTTPS (1)

Encryption works by public and private keys. The public key encrypts the plain text into a cipher text while the private key decrypts from the cipher text into the plain text. (Figure 6.)

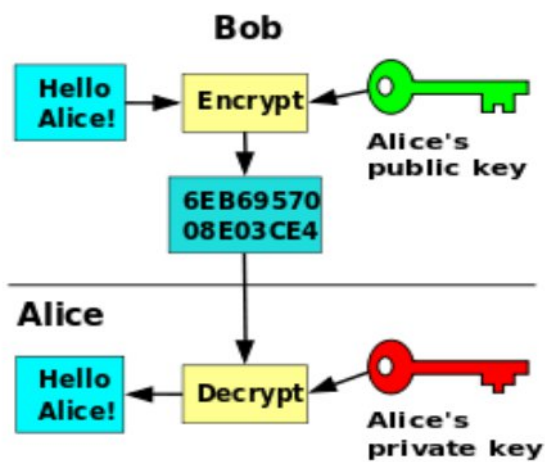


FIGURE 6. Encryption and decryption of data using public and private keys (1)

4 APPLICATION INTERFACE

4.1 Client's interface

In information technology, the user interface (UI) is everything that is designed into an information device with which a human being may interact -- including display a screen, a keyboard, a mouse, a light pen, the appearance of a desktop, illuminated characters, help messages, and how an application program or a web site invites interaction and responds to it. One of the most important aspects of a program is its interface, for a good interface allows the user to interact with the program quickly, easily, and without confusion (8).

Everything stems from knowing your users, including understanding their goals, skills, preferences, and tendencies. Once the user is defined, the best practices of interface design must be considered. The basic practices are:

- **Keep the interface simple.** The best interfaces are almost invisible to the user. They avoid unnecessary elements and are clear in the language they use on labels and in messaging.
- **Create consistency and use common UI elements.** By using common elements in your UI, users feel more comfortable and are able to get things done more quickly. It is also important to create patterns in language, layout and design throughout the site to help to facilitate efficiency. Once a user learns how to do something, they should be able to transfer that skill to other parts of the site.
- **Be purposeful in page layout.** Consider the spatial relationships between items on the page and structure the page based on importance. Careful placement of items can help to draw attention to the most important pieces of information and can aid scanning and readability (9).

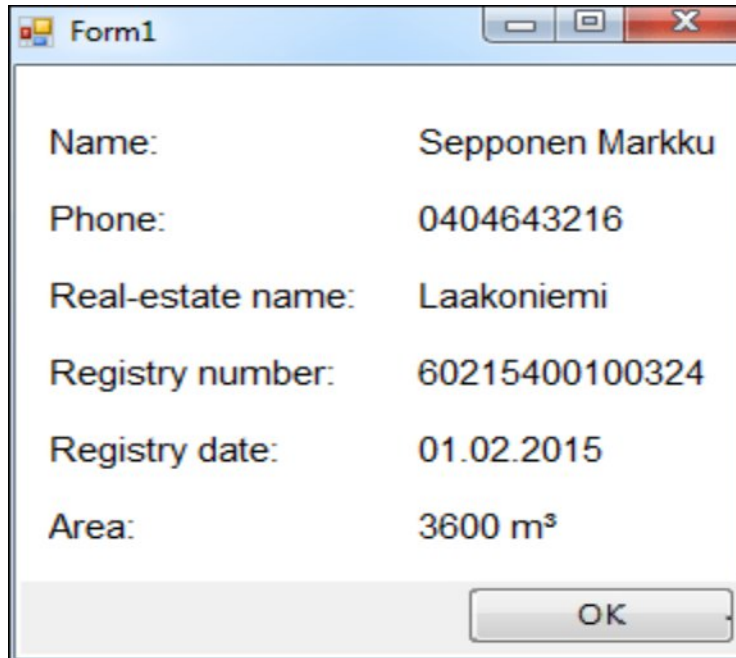
- **Strategically use color and texture.** You can direct attention toward or re-direct attention away from items using color, light, contrast, and texture to your advantage (9).
- **Use typography to create hierarchy and clarity.** Carefully consider how you use typeface. Different sizes, fonts, and arrangement of the text to help increase scan ability, legibility and readability (9).
- **Make sure that the system communicates what is happening.** Always inform your users of location, actions, changes in state, or errors. The use of various UI elements to communicate the status and, if necessary, the next steps can reduce frustration for your user (9).
- **Think about the defaults.** By carefully thinking about and anticipating the goals people bring to your site, you can create defaults that reduce the burden on the user. This becomes particularly important when it comes to form a design where you might have an opportunity to have some fields pre-chosen or filled out (9).

The basic Interface elements are:

1. **Input Controls:** buttons, text fields, checkboxes, radio buttons, dropdown lists, list boxes, toggles.
2. **Navigational Components:** breadcrumb, slider, search field, pagination, slider, tags, icons.
3. **Informational Components:** tooltips, icons, progress bar, notifications, message boxes.
4. **Containers:** accordion

What is basic on the MML interface, the response is provided as an XML document. Thus creating a user control is a good practice to present the data to the client using basic inputs, such as labels and informational component as MessageBox. Adding navigational components is efficient by easing the user interface so that the user does not need to click several times in order to receive the

data. Using a pop-up timer is a good idea, which shows automatically the data to the user. (Figure 8.)



The image shows a screenshot of a Windows-style window titled "Form1". The window contains a list of property details in a simple key-value format. The details are: Name: Sepponen Markku, Phone: 0404643216, Real-estate name: Laakoniemi, Registry number: 60215400100324, Registry date: 01.02.2015, and Area: 3600 m³. At the bottom right of the form is an "OK" button.

Name:	Sepponen Markku
Phone:	0404643216
Real-estate name:	Laakoniemi
Registry number:	60215400100324
Registry date:	01.02.2015
Area:	3600 m ³

FIGURE 8. User interface form where data is presented

4.2 Invoicing

An invoice, bill or tab is a commercial document issued by a seller to a buyer, relating to a sales transaction and indicating the products, quantities, and agreed prices for products or services the seller has provided the buyer (10).

A typical invoice may contain:

- Word *invoice* (or *Tax Invoice*).
- Unique reference number (e.g ID).
- Date of invoice.
- Credit terms.
- Tax payments, if relevant (e.g., GST or VAT).
- Name and contact details of the seller.

- Tax or company registration details of the seller, Name and contact details of the buyer.
- Date when the goods or service was sent or delivered.
- Description of the product(s).
- Unit price(s) of the product(s), if relevant.

The application is using by several customers in which invoicing is implemented based on the customer's identification in the customer system (Figure 9).

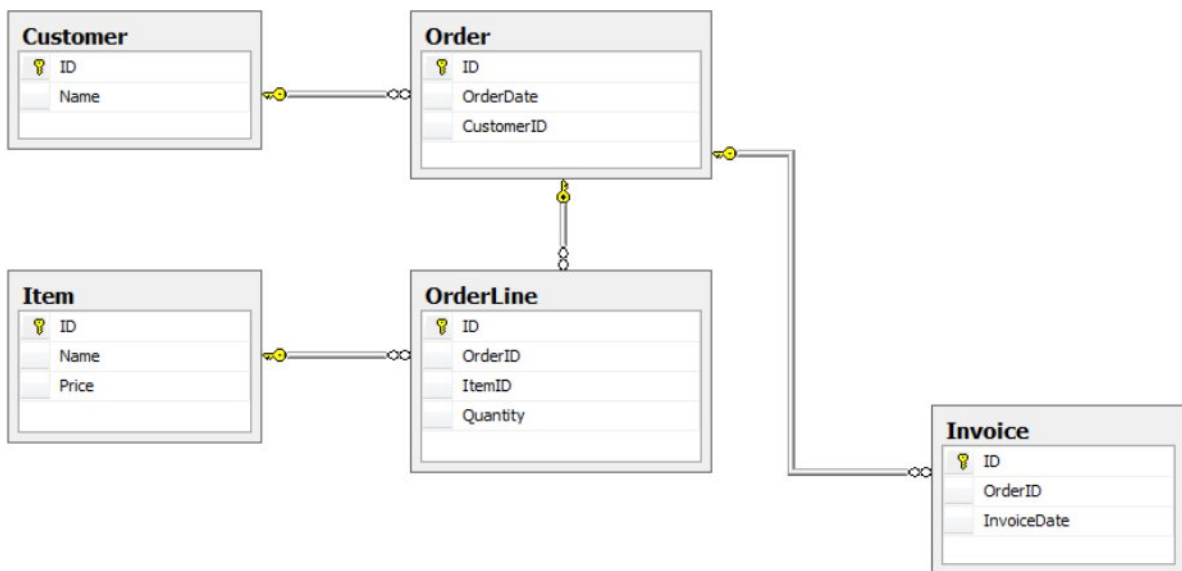


FIGURE 9. General class diagram of invoicing in SQL

5 SUMMARY

The aim of this thesis was to find the most efficient solution for an RE interface. The application was implemented in an independent program (OVT), which was linked into the client's application to work together. The application was supposed to be easy and fast to use, as well as having low cost for the clients. The objectives were achieved and resulted in success due to the user's testing. The RE interface could be developed in the future for a better usability and efficiency.

The topic of the thesis was very interesting since it was implementing a real product for the customers in order to achieve a business. I liked every work step in my thesis so It taught me how to combine ideas and technologies together. I did not have any practical experiences before of APIs and data flow between clouds. I faced many difficulties in drawing the work plan and understanding the the system due to the lack of knowledge. During the implementation the plan was modified several times to achieve more quality results of a better usability for the customers. The MML service was quite challenging to be implemented into Visual Basic because of the data format that MML provides for the developers.

I worked in IT industry with professionals and I used the industry leading development tools. Thus, I have learned how to use new development tools. IT management is one of the greatest skills I have learned in my thesis. I was able to take an advantage of how to draw a business IT plan, how to manage a project and how to implement a real business into an application. Moreover, I have also taken a good understanding of the communication between the customer and the developer, thus getting a better and clearer background image of the IT career. The working environment also had a great influence on my thesis since the general atmosphere of the work environment was quite comfortable for the developer to work, i.e. the availability of the development tools and premises. Also, colleagues were quite supportive when I was facing difficulties in the work.

REFERENCES

1. Google Images / Web . Date of retrieval 11.04.2016
www.Google.fi
2. InstantSSL. Date of retrieval 16.04.2016
www.instantssl.com/
3. MSDN visual basic documentation. Date of retrieval 29.02.2016
[https://msdn.microsoft.com/fi-fi/library/456dfw4f\(v=vs.110\).aspx](https://msdn.microsoft.com/fi-fi/library/456dfw4f(v=vs.110).aspx)
4. National Land Survey MML. Date of retrieval 11.04.2016
<http://kansalaisen.karttapaikka.fi/koordinaatit/koordinaatit.html>
5. Project smart, project planning. Date of retrieval 07.04.2016
<https://www.projectsart.co.uk/project-planning-step-by-step.php>
6. Request/Response documentation. Date of retrieval 25.03.2016
<http://rve.org.uk/dumprequest>
7. Techopedia. Date of retrieval 16.04.2016
<https://www.techopedia.com/>
8. Techtarget, Introduction to interface. Date of retrieval 19.04.2016
www.techtarget.com/
9. Usability. Date of retrieval 20.04.2016
<http://www.usability.gov/>
10. WikiPedia. Date of retrieval 20.04.2016
<https://en.wikipedia.org/>
11. XML Extensible Markup Language. Date of retrieval 27.03.2016
<http://vb.net-informations.com/xml/vb.net-xml-tutorial.htm>

