

Mikias Mulugeta

Distributed Database Integration with Web Services

Helsinki Metropolia University of Applied Sciences
Bachelor of Engineering

Media Engineering

Thesis

5 May 2011

| | |
|--|---|
| Author Title | Mikias Mulugeta Distributed database integration with web services |
| Number of Pages Date | 46 pages 5 May 2011 |
| Degree | Bachelor of Engineering |
| Degree Programme | Media Engineering |
| Specialisation option | Advanced application development |
| Instructor(s) | Eskindir Abdela, Project Manager Kari Aaltonen, Senior Lecturer |
| <p>The main purpose of the thesis project was to build an application that has a reservation system by accessing distributed database remotely. There are many parts of the world where the ICT technologies are growing at a higher rate than the medical and transportation infrastructures. The electronic pharmacy application was intended to take advantage of this reality by accessing inventory databases of pharmacies from anywhere in the world via web service and make medicine reservation from a pharmacy located nearest to the patient.</p> <p>The application was made by Microsoft tools and programming frameworks. Never the less, the thesis study includes the general structure of using web service and how database can be accessed and manipulated remotely. The E-pharmacy application includes authentication of users with different roles, manipulation of data, reservation system and mapping of locations.</p> <p>As companies' implementation of databases increases these days, the use of service oriented applications that use these databases remotely also increases. Web services are the standard for building service oriented applications. Building a web service requires choosing the right type of bindings based on the connection types to choose, hosting the service on the right type of environment, setting the address that exposes the service and considering security issues.</p> | |
| Keywords | web service, database, Siveright, mapping |

| | | |
|-----|--|----|
| 1 | Introduction | 3 |
| 2 | Web services and data repository | 4 |
| 2.1 | Types of web services | 4 |
| 2.2 | Web service technology stack | 5 |
| 2.3 | Simple Object Access Protocol (SOAP) | 6 |
| 2.4 | Database web service | 7 |
| 3 | Web service frameworks | 9 |
| 3.1 | Communication contracts | 10 |
| 3.2 | Setting data bindings | 12 |
| 3.3 | Addresses for end points | 14 |
| 3.4 | Client applications accessing the service | 15 |
| 3.5 | Hosting services | 17 |
| 3.6 | Web service security issues | 18 |
| 3.7 | Bing map | 20 |
| 4 | Silverlight | 21 |
| 4.1 | Silverlight roadmap | 21 |
| 4.2 | User interface controls | 23 |
| 4.3 | Windows communication foundation data services | 27 |
| 5 | Implementation of LiveMed electronic pharmacy | 29 |
| 5.1 | Requirement | 29 |
| 5.2 | Use cases | 30 |
| 5.3 | Database | 31 |
| 5.4 | Development steps | 32 |
| 5.5 | Authentication and registration | 35 |
| 5.6 | Data handling | 36 |
| 5.7 | Reservation | 37 |
| 5.8 | Mapping implementation | 39 |
| 5.9 | Publishing | 41 |
| 6 | Conclusion | 43 |
| | References | 45 |

1 Introduction

The obstacles of giving a proper medical service are paramount. Acquiring prescribed medicines at the right time and knowing where exactly to find them are among the many challenges patients face in many parts of the world. Especially in developing countries, hospitals do not have much communication with pharmacies unless they are under the same ownership or administration. When a patient gets a prescription he/she has to travel to a pharmacy to see if the medicine is available. One can imagine the amount of time and energy the patient wastes which may be very essential in saving his/her life. The transportation cost is also an important issue.

The thesis project I completed for LiveMed Company addresses this issue by providing a platform where pharmacies expose their inventory medicine database to a web service, to give an access for client applications that may be located in any part of the world. LiveMed is a company that builds software applications for health infrastructures. It has already a Tele-medical portal system that consists of scheduling system and video conferencing.

The thesis project has primary and secondary objectives. The primary goal is to build an application with Microsoft development tools and programming frameworks, a single pharmacy database, a host application that hosts the database and a Silverlight client application that connects with the database and makes reservation. The secondary goal was to add extra functionalities to the application that manipulates the data exposed by the web service, and use different components of Silverlight to make the electronic pharmacy look a complete application.

As systems become more service oriented, the significance of connecting services to clients consuming those services rises. The approach I chose in building an application that requires accessing of distributed databases remotely is web services. In this thesis, I will explain the main aspects of web services and how it is possible to use them for accessing and manipulation of databases. From the multiple web service programming frameworks currently available, I chose Windows communication foundation in building LiveMed electronic pharmacy.

2 Web services and data repository

Web services are a set of protocols and standards used to exchange data between applications. Various software applications developed in different programming languages and executed on any platform, can use web services to exchange data over computer networks like the Internet. A web service is a set of self describing self contained software module that can be accessed via network. It can be used for the purposes of solving problems, complete tasks or conduct transactions on behalf of application. The web service interface is built in such a manner that it is independent of the operating system, the programming language used to implement it and the underlying platform. The service can be integrated with remotely located application as it can be accessed programmatically. [1,4-7]

There are different functionalities that web services provide. A web service can be a self contained business task such as fund depositing and fund withdrawal; it can be a full-fledged business process such as automated purchasing of office supplies; it can be an application such as air lines flight reservation application or it can also be service-enabled resource such as accessing a database that contains medical record of patients. [1,6]

2.1 Types of web services

Based on the protocols they use there are two types of web services. These are simple or informational services and complex services or business process.

Simple or informational services; these are simple by nature and provide access to content interacting with end user or by exposing back end business applications to other applications. The programmatic services are the services that expose the business functionality of the application and the components that underlie them. According to the business problems they solve, simple services can be divided into three categories. These are pure content service, simple trading service and informational syndication service.

Pure content service gives programmatic access to content such as weather report information, news information, design information, simple financial information and etc. Simple trading services are a complex form of informational services that can pro-

vide information across poor systems and information sources including back-end systems giving a programmatic access to a business information system. Information syndication services are value added information web services that assert to plug into commerce sites. [1,14]

Complex service or business process; complex service involves the assembly and the invocation of pre-existing services that can be found in different enterprises to complete a multi step business interaction. When multiple enterprises want to integrate their business logic together like ordering, logical support and technical support; they need to integrate several services together. The way to do that is by using complex web services.

In complex services that compose programmatic web services, the client can assemble the simple services to make the complex service. An example for this scenario can be an inventory checking service that consists of a part of an inventory management process. The complex service that composes interactive web services exposes the functionality of the applications web presentation or browser layer. [1,14-16]

2.2 Web service technology stack

Web service architecture involves many layers and technologies that consist of different standards. At the transport layer it uses different protocols such as HTTP, JMS and SMTP. XML is the fundamental block for every layer. At the base line SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) and UDDI (Universal Description, Discovery and Integration) technologies are used. SOAP is used for exchanging structured information of web services; WSDL describes the service as collection of network end points or ports; UDDI is XML based registry mechanism to register and locate web service applications.

Figure 1 shows technologies used at different layers of multiple kinds of web services.



Figure 1. Web service technology stack [2,38]

The web service stack also holds other collaborating standards other than the two mentioned above in the transport layer and the core base line standard. They are service composition standards, service collaboration standards, transaction standards and value added standards. Service composition standard engage in defining the rules for managing data. And this can be materialized by the business Execution language (BPEL). Service collaboration standard is materialized by the web services choreography description language WS-CDL. Applications that use transactional system or work flow system implement transaction standards. Value added standards provide security and authentication mechanism. [2, 36-41]

2.3 Simple Object Access Protocol (SOAP)

Simple object access protocol (SOAP) is an XML based communication protocol for accessing a Web service. A better way for application to communicate via browser is by using Http as it supported by all browsers and servers. To facilitate this functionality SOAP is used. It provides a means to communicate between applications that use dif-

ferent programming languages, running on different operating systems and that use different technologies. Figure 2 shows the communication between service requester and service provider by using SOAP messages.

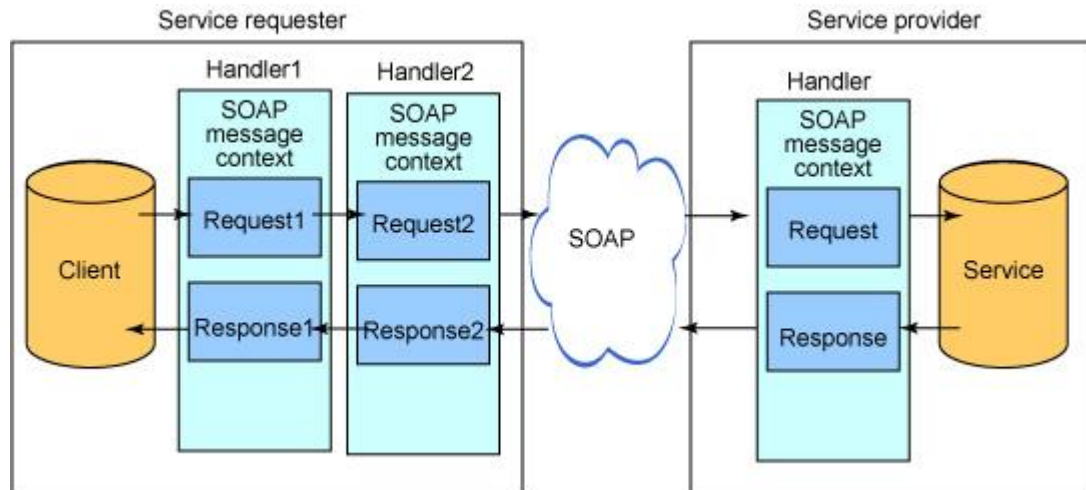


Figure 2. SOAP message flow diagram [3]

Elements contained in SOAP message are an envelope element that identifies the XML document as a SOAP message, a header element that contains header information, a body element that contains call and response information and a fault element that holds errors and status information. [4]

2.4 Database web service

Almost every industry in the world possesses databases. Database web services technology is a database approach that provides access to access data and Meta data through the web services interface. The Database web service follows two directions. These are accessing database source as a web service and consuming external web service from the database itself.

While accessing the database client queries can be made to retrieve data from database and invoke procedures using the standard web service protocols. The client applications can reside in different location and platform can also use any kind of data format and implementation. The client calls the database first using the database as a service provider. This allows control of the existing SQL, procedures and classes within the database. Then the database table can be accessed and altered by the client.

Figure 3 shows a generic database web service frame work where it uses a file based approach for designating the database operations that are to be exposed to the web service. [5,199-202]

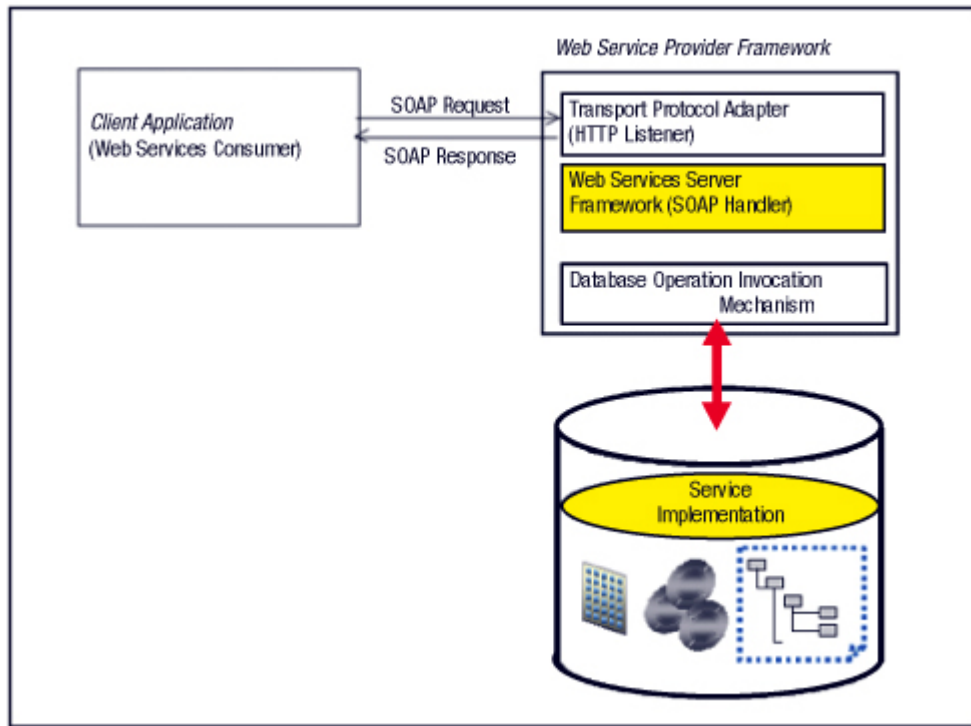


Figure 3. Generic database web service frame work [6]

While using database as a service consumer, relational database storage, indexing and searching capabilities can be included in non structured data including web service. Dynamic data such as currency change data can be tracked, refreshed and be handled for other functionalities by calling a web service. [5,199-205]

3 Web service frameworks

For multiple kinds of programming languages there are many web service programming frameworks. Each has their own specific properties in the kind protocols they use, the service they grant and the messaging model or destination they communicate. Out of these frameworks I chose windows communication foundation specifically to explain the detailed process of using web services.

WCF is a framework used to design service oriented applications through its simplified model. It allows developers to create diverse ranges of applications which is particularly suited to XML based web standards. The most notable advantage of using WCF is its capability to integrate multiple technologies at the same time. Today's computing involves many distributed technologies which each has its own specific role and may be applicable in a different programming model. It is very difficult to "plug and play" distributed APIs without authoring a considerable amount of custom infrastructure [1]. For example, if you build your system using the .NET remoting APIs, and you later decide that XML web services are a more appropriate solution, you need to reengineer your code base. [7]

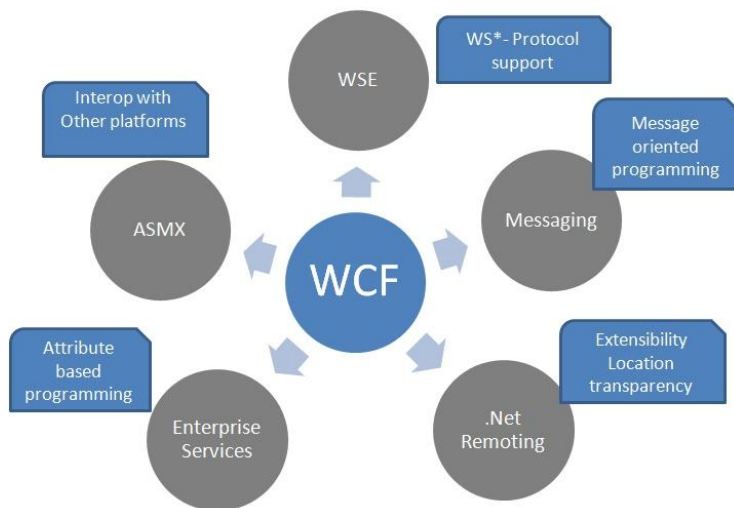


Figure 4. WCF model [8]

A developer using WCF has many distributed technologies under the same API and all that is expected to do is use the reference System.ServiceModel assembly and import

its namespaces. Figure 4 shows the distributed technologies included by web communication. [9,882]

3.1 Communication contracts

During the communication of two systems over a network, the systems need to agree on what kind of information is sent back and forth. WCF also formalize this and the term for it is called contracts. Therefore a contract is a platform independent and standard way to describe what the service can do. In WCF there are four types of contracts.

Service contract

A method used defining the interface to allow the WCF client understand what they can do with the service. The code below shows one service contract used in a WCF application.

```
{
[ServiceContract]
public interface ILiveMed
{
    #region Countries

    [OperationContract]
    List<Country> GetCountryList();
}
}
```

Figure 5. Service contract

In this example, the service contract that the service class would implement is ILiveMed. For each method that the WCF client accesses there is an operation contract that must be added.

Data Contract

Data contract defines what types of data are transferred in the service. Without including this contract there certain types allowed to be passed by default. These are basic data types like int, String, DateTime and XmlElement. However, new complex data types created need to have their own data contract. For custom data types like custom class, a data contract needs to be defined. There are two types of data contracts that

are named as Data contract and Data member. While the former is an attribute defining the class the later is an attribute used to define properties. Data contracts are required if the service contract defines operations that return instances of complex types. [8]

Fault Contract

The service that is developed in this study has a run time error. This error should be reported to the client in a proper manner. Basically when a managed application or service is developed, we will handle the exception using try- catch block. But these exception handlings are technology specific. Hence, not all exceptions thrown from the service reach the client side. WCF provides the option to handle and convey the error message to client from service using SOAP Fault contract. Suppose the service I consumed is not working in the client application. I want to know the real cause of the problem. How can I know the error? Fault Contract provides documented view for an error occurred in the service to client. This helps the client to easily identify the run time errors. [8]

Message contract

Most of the time, the developer will concentrate more on developing the DataContract. WCF will automatically take care of the message. Sometimes the developer will also require control over the SOAP message format. In that case WCF provides Message Contract to customize the message as per requirement. Integrating with another non-WCF service require using the MessageContract, MessageHeader, and MessageBody attributes. [10,394-396]

In Figure 6, it is shown how Message contract is defined and used as a parameter in a service operation.

```
[MessageContract]
public class UserMessage
{
    [MessageHeader]
    public int UserId;
    [MessageBody]
    public UserData User;
}
[ServiceContract()]
interface ILiveMed
{
    [OperationContract()]
    void SaveUser(UserMessage UserMsg);
}
```

Figure 6. Message contract

3.2 Setting data bindings

Once the WCF contracts are created and the service implementation is done, data binding information must be specified so that the WCF client can access the information. Binding is a group of binding elements; each element is responsible for different things, such as message encoding or security. WCF bindings can specify certain characteristics such as; the contracts implemented by the service, the transport layer used to move data (HTTP, MSMQ, named pipes, TCP), the channels used by the transport (one-way, request-reply, duplex), and the encoding mechanism used to deal with the data itself (XML, binary, etc.). the information provided by the WCF bindings include supported web service protocols (if permitted by the binding) such as WS-Security, WSTransactions, WS-Reliability, and so on. [1]

The fact that WCF comes with a number of pre-configured bindings makes WCF development peculiar from the .NET remoting and/or XML web service development. These pre-configured bindings bring groups of binding elements together to address common scenarios that developers need. Each channel within the binding is part of the overall communication. This allows for re-use within WCF and allows you to create your own binding using pre-configured elements that already exist. You can use one of the many pre-defined bindings provided for you by the WCF framework, which try to address many different scenarios that are commonly used at the time. [10,385]

If the binding that is provided by the framework doesn't quite address your specific needs, you have a few options. First you can tweak one of the pre-defined bindings that closely match your needs by setting one of the many binding properties. You can also define your own binding by selecting existing binding elements and creating a group of binding elements. For the most part, you will be able to use one of the default WCF bindings. This speeds up development time immensely, as you do not have to write code to transport messages over the wire. You are able to define the ABCs and then everything is taken care of for you. Due to having these binding implementations set for you by WCF, you can have multiple endpoints with different bindings. So your same service method could implement several different protocols and transport options. This allows you to quickly expose your service to any kind of client that needs to access it. You could use a .NET specific binding to communicate with .NET clients to take advantage of the built-in WCF security and performance capabilities. However, you can also expose your service using WS-standards for Java, PHP, or any compatible clients that implement the specifications. The example below in figure 7 shows the service binding used in the LiveMed Electronic pharmacy application. [6,482-486]

```
<configuration>
<system.serviceModel>
<bindings>
<basicHttpBinding>
<bindingname="BasicHttpBinding_ILiveMed" maxBufferSize="2147483647"
maxReceivedMessageSize="2147483647">
<security mode="None" />
</binding>
</basicHttpBinding>
</bindings>
<client>
<endpointaddress="http://localhost:1430/LiveMed.svc"binding="basicHttpBinding"
bindingConfiguration="BasicHttpBinding_ILiveMed"contract="ServiceReference.ILiveMed"
name="BasicHttpBinding_ILiveMed" />
</client>
</system.serviceModel>
</configuration>
```

Figure 7. Web configuration for service binding

Binding types

Http bindings - The BasicHttpBinding, WSHttpBinding, WSDualHttpBinding, and WSFederationHttpBinding options are geared toward exposing contract types via XML web service protocols. For projects that require multiple operating systems and multiple programming, Http binding is the preferred option as it uses XML for encoding data and uses Http on the wire. [11,881-882]

TCP binding- Is applicable on distributed application running on machines that have .Net 3 or more configured on them. When using this type of binding both the host and the client must be .Net application. In the cases where both Web service binding and TCP binding are used together, the performance is enhanced as it ensures all data is encoded in a compact binary format rather than XML. The NetTcpBinding class uses TCP to move binary data between the client and WCF service. This will result in higher performance than the web service protocols. NetNamedPipeBinding supports transactions, reliable sessions, and secure communications, but it has no ability to make cross-machine calls. For the fastest way to push data between WCF applications on the same machine, NetNamedPipeBinding is the better binding choice. [11,882-883]

MSMQ-based binding- to integrate with a Microsoft MSMQ server, the NetMsmqBinding and MsmqIntegrationBinding bindings are used. The NetMsmqBinding is used to enable WCF applications to send and receive messages to and from existing MSMQ applications that use COM, native C++, or the types defined in the System. MsmqIntegrationBinding is used for cross-machine communication between .NET applications. [11, 884]

3.3 Addresses for end points

Each end point should have unique address as it is the means to locate the service. An address can be any kind of address such as the IP address, URL and server name. The same way as URL serves the purpose in website, and a service needs to have an address to know where it is hosted. An address is composed of Transport scheme, server location, port and path. Transport scheme holds information on the beginning part of the address that sets the protocol the service uses. Server location has information on the actual location of the server where the service is hosted. This can be on local network or across the web. Port information is usually set by default but it can also be

specified. Path is the relative location of the resource on the server. If the resource file name is not specified the server returns a default value which the format of the message would be scheme://SERVERLOCATION[:port]/path/subpath.[12,70]

```
<endpointaddress="http://localhost:1430/LiveMed.svc"
binding="basicHttpBinding"
bindingConfiguration="BasicHttpBinding_ILiveMed"contract="ServiceReference.ILiveMed"
    name="BasicHttpBinding_ILiveMed" />
</client>
```

Figure 7. WCF address

In the figure 7 the example shows the client gets the address of the service with the end point address specified. The table below illustrates possible end point addresses and the corresponding transport protocols used.

Table 1. WCF address examples per protocol [12,71]

| Transport protocol | Example Address |
|--|---|
| Http | http://localhost:8001 http://localhost:8001/Service1 |
| Http (secure) | https://localhost:8001 |
| TCP | net.tcp://localhost:8002/Service1 |
| Peer network | net.p2p://localhost/ |
| IPC (Inter - process communication over named pipes) | net.pipe://localhost/PipeService1 |
| MSMQ (Microsoft Message Queue) | net.msmq://localhost |

As seen in the table, the end point address has 4 parts. The scheme (http), the address of the machine, the port number and the exact path of the service.

3.4 Client applications accessing the service

When the service completed, there have to be a client application that uses the service. While it is possible to code everything from scratch, .Net 4.0 provides the necessary approaches to generate a client side proxy. If we for instance consider building a console client application, under the right click options on the solution of the project

we have an option which says service reference. We use dialog box to connect our application with the service. As seen in figure 8, the Add reference dialog has a Discovery button which provides a means to discover the already existing services in the solution. [9,493-495]

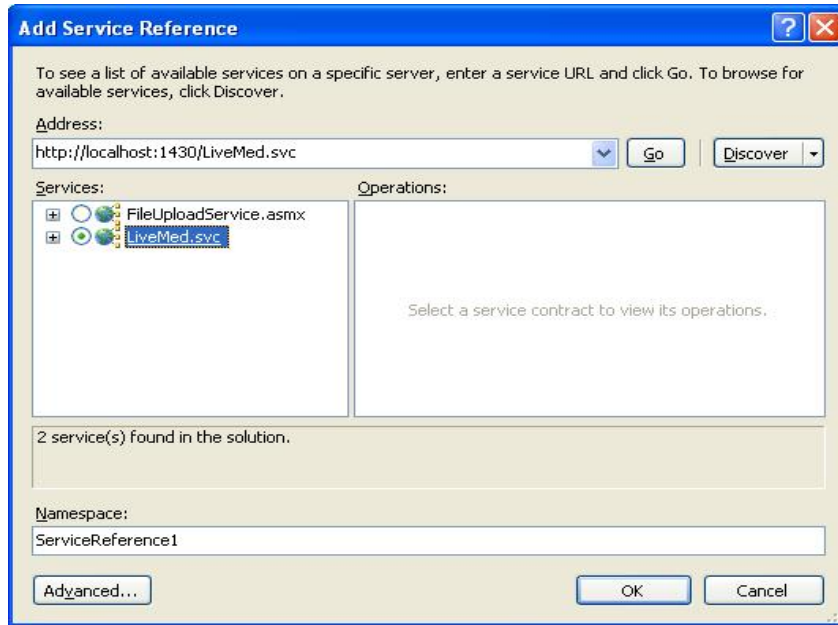


Figure 8. Add service reference wizard

While using visual studio to be able to connect with the hosted service, the service needs to be up and running before we open the Add Service Reference dialog.

Figure 9 shows how Visual Studio presents the added service.

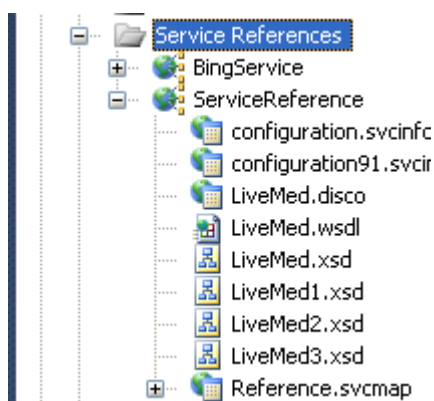


Figure 9. Generated files in a service reference

All the lists of services, contracts and operations are useful in identifying the exact type of service the client application needs to be connected to. After we complete choosing our service and connect it with the client, visual studio generates some code when adding the service reference.

3.5 Hosting services

Many of the services inside ASP.NET web applications, Windows Services, console applications, or even applications with GUIs built with Windows Forms or WPF can be hosted and can be flexible about their location. The only place where WCF service can not be hosted is in a process that has security constraints set to the network such as web browsers. For instance a Silverlight application can only use a web service as a client. It cannot host the service. [9,486-489]

WCF service can be hosted using IIS, WAS (Windows activation service) and self hosting. These hosting environments expose the service to client applications. Some of the features considered accountable in making a host environment a great one are;

- Executable Process/Application Domain: implies the existence of an application domain.
- Configuration: to support deployment and managing.
- Activation: to initialize the channel stack that receives incoming messages.
- Idle-time management: to conserve resources during idle time.
- Health mentoring: to ensure availability of the host.
- Process Recycling: To avoid problems associated with memory leaks or faulty code.
- Management Tools: tools for configuring hosting features for greater control and manageability.

When using IIS what needs to be done in the setting is creating virtual directory and on the virtual directory the location of the service has to be pointed. A service hosted by IIS is launched automatically upon the first client request. What makes IIS disadvantageous is the fact that it uses only HTTP protocol. [13;14,388]

WAS is a process activation service installed in IIS 7.0 version. The IIS 7.0 version is found in Windows vista operating system and the latest versions. This hosting environment is different from IIS for supporting other protocols than HTTP such as TCP

and MSMQ. Just like IIS, WAS provides features for idle-time management, health monitoring, recycling and management tools. [13]

Self Hosting requires adding managed codes to the service and this code can either windows form application or console application. In self hosting the host process needs to be started and running before the client makes a call to the service. The host application must specifically create and open an instance of ServiceHost object. Then the service host remains open and available.

While using all the hosting types, information about the hosting type is specified so that the world can access it. [13]

```
<system.serviceModel>
  <behaviors>
    <serviceBehaviors>
      <behavior name="">
        <serviceMetadata httpGetEnabled="true" />
        <serviceDebug includeExceptionDetailInFaults="false" />
      </behavior>
    </serviceBehaviors>
  </behaviors>
  <serviceHostingEnvironment multipleSiteBindingsEnabled="true" />
</system.serviceModel>
```

Figure 10. Service file

The example in figure 10 demonstrates the config file that allows the service to be accessed. The first section notifies that it is possible for the service to publish Meta data information and the second section provides information about the current hosting environment.

3.6 Web service security issues

WCF is a soap communication based distributed platform. Since many clients can use the same service, as any distributed system required it, securing the message between the service and the client is very imperative. WCF provides a rich and configurable environment for creating security policies and to control them. Numerous multi-

authentication mechanisms are included such as windows token and username and password, user tokens and certificates. The windows role setting, ASP role, and custom authorization methods can be used to assign roles for different user groups. The other important elements for security, integrity and confidentiality can be based on symmetric session keys, or asymmetric keys for single-hop protection. The primary step in setting WCF security is defining the security policy. This includes setting requirements for authorization, authentication and message protection so later can be imposed with service configuration.

The type of binding used and behaviours influence the configurations options for service security policy. All bindings have five security modes. These are;

- None: turns security off entirely or allocates authentication between transport and message level security
- Transport: each transport protocol has its own way for message credentials and managing message protection.
- Message: uses message security for mutual authentication and message security.
- Both: allows supplying settings for transport and message level settings.
- TransportWithMessageCredential: Credentials are passed with the message and message protection and server authentication are provided by the transport layer.
- TransportCredentialOnly: Client credentials are passed with the transport layer and no message protection is applied.

When security mode is set to manage security like in the case of Net TCP binding which is secure by default, the default security settings can be customized.

The other setting important in WCF programming is setting client credentials. From the available credential types windows, certificate, digest, basic, username NTML and issued token the appropriate credential type must be chosen depending on the security mode already set. [7;13]

3.7 Bing map

Bing map is "an online mapping service that enables users to search, discover, explore, plan, and share information about specific locations" [15]. Bing map provide road view, Arial view, bird's-eye view, 3D maps and street side view for many locations worldwide. Bing map provide an opportunity for developers to merge location and local search feature with their own application.

Bing map implementation

Bing map implementation requires four steps. The first step is to create a Bing maps Developer Account. In order to use the Bing Maps Silverlight Control, you need a Bing Maps Key to authenticate your application. To obtain Bing Maps Keys you should first go to "<https://www.bingmapsportal.com>" and create a Bing Maps Developer Account. Then create a key to access your application. [15]

The second step is downloading the Bing map Silverlight control. It is available on the official Microsoft Silverlight web site. This download includes the Bing maps Silverlight control assemblies and an offline version of the Bing maps Silverlight control SDK. After downloading, this control needs to be installed on the system. [15]

The third step is to use control in application. When Bing map Silverlight control is installed, it will have related libraries located in directory C:\Program Files\Bing Maps Silverlight Control\V1\Libraries. The Silverlight project needs to add the reference of Microsoft.Maps.MapControl.Common.dll and Microsoft.Maps.MapControl.dll. [15]

The final step is to add web reference of GeocodeService Bing map web service. Bing map control directly does not take address to populate location on graph. Bing control takes latitude and longitude to populate the map. This web service takes the address of a location and gives latitude and longitude of that location. Then latitude and longitude are given to Bing map control to populate the location in the map. [15]

4 Silverlight

Microsoft Silverlight is a “cross-browser, cross-platform implementation of the .NET Framework for building and delivering the next generation of media experiences and rich interactive applications (RIA) for the Web [16].” It can also be used to build desktop applications and windows mobile applications. Silverlight uses XAML (Extensible Application Markup Language) for UI generation and managed code for application logic. The Silverlight browser plug-In is very light and supports Mozilla Firefox, Google Chrome, Apple safari, Microsoft Internet Explorer, and Apple Mac OS X browsers. Also it does not need any browser specific coding in order to make Silverlight applications work on multiple browsers. [7;16]

What makes Silverlight advantageous for developers is the fact that it combines many features into one development platform. Some of the features Silverlight platform includes are multiple browser support such as extensions to JavaScript, integration with existing applications, access to .NET programming model, development tools support, networking support, LINQ (language integrated query), WPF and XAML. [16]

4.1 Silverlight roadmap

Before the first version of Silverlight came out, it was named as WPF/E (Windows Presentation Foundation) which was solely used for Silverlight application. Hence, the prime intention of making Silverlight platform was to make an advanced version of WPF compatible with browsers. [16;18,16]

Silverlight 1.0

When this version came out in March 2007, it did not have any .NET support and uses XAML or a mix XAML and JavaScript. It has basic layout which the canvas was the only layout component. It is also limited to basic controls which were only the TextBlock and Run control for displaying text. Even though 2D graphics is included the Geometry based classes can't be directly placed on the user interface. Media support for videos and images is available in this version that is effective in downloading easily making the best use of bandwidth available. Animation and brushes & transformation became available on the first version of Silverlight. [16; 18,16]

Silverlight 2.0

The second version which was named Silverlight 1.1 at the time due to fast transformation from the first version was revolutionary as it brought the .Net framework integration for the first time. It also included standard control set and an input event system which used to be handled by developers manually. Enhanced video and many other exciting features were included. [18,16]

Silverlight 3.0

This version came out with extended data layer to create foundation for business oriented scenarios. The media layer was extended to support new formats and new effect known as pixel shaders was introduced. Animation improvement was enhanced with smoother movements closer to reality and 2D elements could be transformed to 3D space. For the first time the OOB (Out Of Browser) feature was seen. Despite the curiosity and admiration of many, it was still left incomplete as it was not granted with any additional permission. [18,16-17; 16]

Silverlight 4.0

This Silverlight came out with many new features. Some of these are Google chrome support, WCF RIA services, printing support, webcam and microphone support, improved mouse support, deep zoom performance enhancement, clip board and drag and drop support, enhanced animation effects and data binding, localization enhancement with bi-directional text, new notification support to display message to end users. [16]

Silverlight 5.0

This is still on beta version which the full version due at the end of 2011. And the new features supported are built-in remote-control support, power awareness, faster application start up, 64-bit browser, Automated UI testing for applications with Visual Studio 2010, improved text clarity, debug data-binding expressions, set breakpoints on bindings, GPU accelerated video decoding, built-in 3D graphics support, and Variable speed playback of media content with automatic audio pitch correction. [16]

4.2 User interface controls

In the Electronic pharmacy project standard types of controls were used that has to do mainly with text inputs, event handling controls and item handling. Hence, I will discuss these controls in this chapter. Expression blend offers variety types of controls and among those the text related controls are TextBlock, TextBox, PasswordBox and TichTextBox.

TextBlock is used to show some information to user or display a label to an input field. It displays a block of a text in its simplest form. If the need arises to change the appearance of the text like the font and color options, it can be done manually from the XAML or from the properties option in the expression blend. The great highlighted part in figure 11 is an example of TextBlock where the label "user" is displayed. [19,83-86]

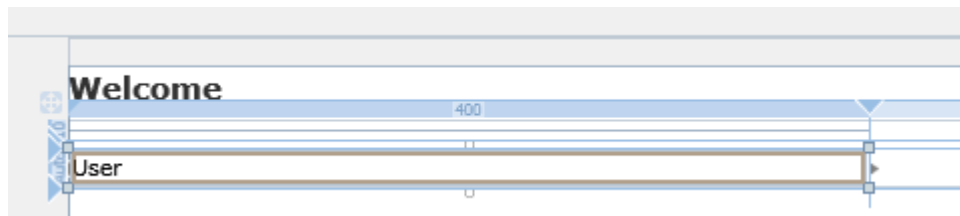


Figure 11. TextBlock

TextBox, RichTextBox and PasswordBox on the other hand allow users input text. In the cases where user is required to fill user name and password, the later type of text control is used. TextBox control in Silverlight is similar to the ones found windows programs and web pages. It also provides functionality to click and drag for text selection, double clicking to select a word and the pressing Ctrl+C to copy and Ctrl+ v to paste for selected text. The selected portion for the user name value for instance is a Text-Box which the user is allowed to input a value. [19,86-89]

Figure 12 illustrates a login page which uses TextBox to fill the values of user name and password.

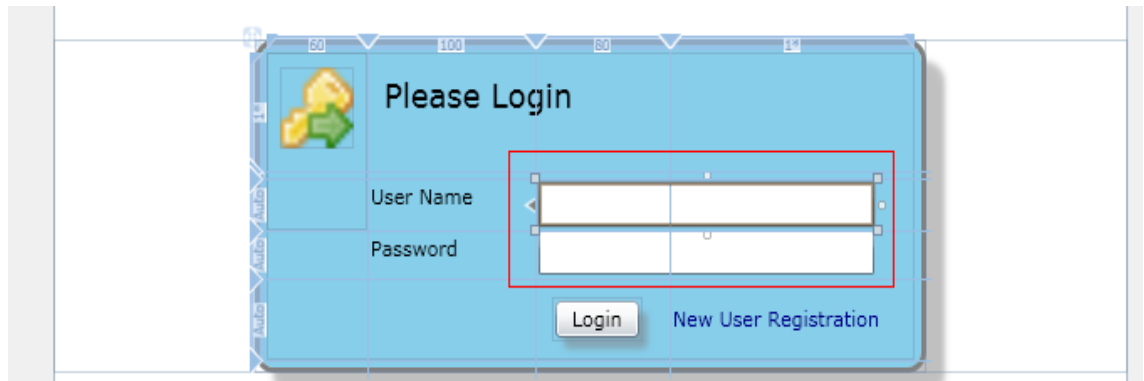


Figure 12. TextBox

The Button control is an event handling control. The text appearing on the button can be altered both manually and using the content property settings. However the button control can have more than a text on it. Unlike TextBox and TextBlock it can also have video, picture and other control and hence why it is called content property instead of text. Figure 13 shows the user interface of a button and its XAML code. [17,27]

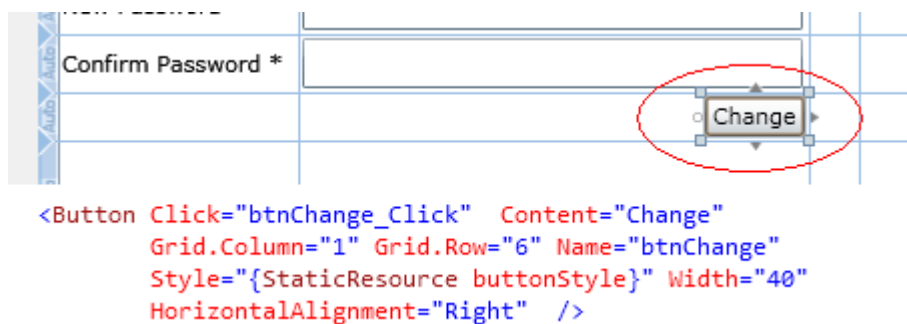


Figure 13. Button control and mark of the button

To add event handler to the button first the button needs to be selected with selection tool and from the property panel Events needs to be clicked which then gives the event field that generates the Button_Click even handler upon double clicking. The event handler can me modified on the code according to functions the button is expected to provide. [19,91-92]

The example below shows the event handler for the button in figure 13. It changes the password of the client.

```
private void btnChange_Click(object sender, RoutedEventArgs e)
{
    if (IsValidate())
    {
        SR.LiveMedClient client = new SR.LiveMedClient();
        client.ChangePasswordCompleted += new EventHandler<SR.ChangePasswordCompletedEventArgs>(client_ChangePasswordCompleted);
        client.ChangePasswordAsync(txtLoginId.Text, txtOldPassword.Password, txtNewPassword.Password);
    }
}
```

Figure 14. Event handler for a button

The items controls are controls that enable display multiple items. These controls are ListBox, ComboBox and TreeView. A ComboBox can help build rich menus that hold more than just texts. It presents list of items where the users has an option to select and view the items by selecting on the control. At a time it displays only one item. To determine the index of the selected item, SelectIndex property is used and if the value of this property appears to be -1, it means there is no selected item. ListBox control is also similar to ComboBox control, however, user is not required invoke an action to view the items. Instead all the items are viewed at once except in the cases where the items are too many for the size of the control. In such a case hidden items are seen by scrolling down. [17,46-50]

In figure 15 the red highlighted part is a ComboBox and the yellow highlighted part is a ListBox where dynamically filled items are presented in the box.

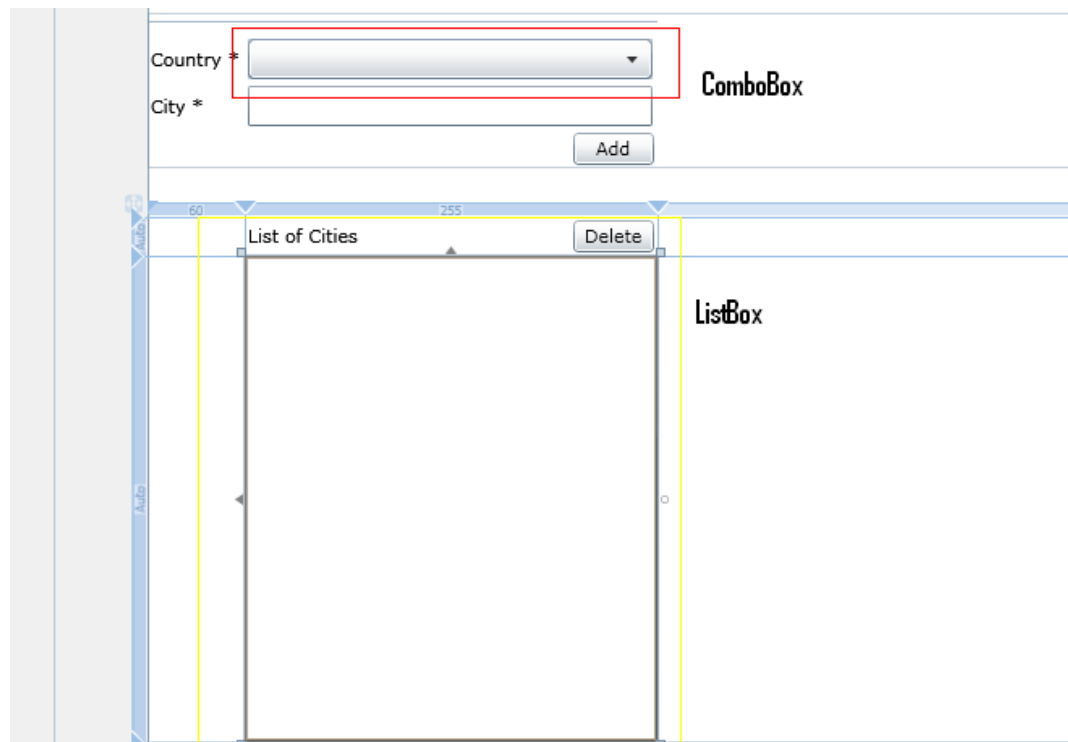


Figure 15. ListBox and ComboBox

Layout containers are controls that can contain other controls. The layout containers used in Silverlight are Grid, Canvas, ScrollViewer, ViewBox, StackPanel, DockPanel and Wrap Panel. In addition to these Border is a common container that basically draws a border around its content. All the the containers have basic properties for positioning and sizing of the container. For instance, properties that help to set the width and height of the containers are the most common ones. Choosing the specific type containers for a Silverlight project needs to be given high consideration beforehand. Containers can be used with in another container and the more we have child containers the harder it is to control the parent container. More complex containers such as Grid take more time to organize than the simple ones. Containers like border, ViewBox and ScrollViewer can only contain one child. [19,107-109]

The canvas control does not provide automatic layout and it is the simplest one. It provides free space for the developer to paint anywhere in the field. The Canvas left and the Canvas top properties enable set the layout. [17,53-55]

Using grid control works in such a way that the screen is divided into invisible grid of cells which individually can have their own set of heights and width. The columns in grid control have three properties that specify their minimum, maximum and current width. Grid control also has resizing property that help the application respond quickly as soon user is resizing it. In figure 16 the grid container is seen divided in to grid cells which each contain different controls inside them. [17,53-59]

| User | |
|--|------------------------------|
| General Detail | |
| Title * | First Name * |
| Middle Name | Last Name * |
| Country * | City * |
| Working Hospital | Address |
| Phone | Mobile |
| Email * | Is Super |
| Login Detail | |
| Login ID * | Is Active |
| Password * | Valid Upto * <dd/MM/yyyy> 15 |
| Confirm Password * | |
| <input type="button" value="New"/> <input type="button" value="Save"/> <input type="button" value="List"/> | |

Figure 16. Grid control

StackPanel layout enables to create horizontal and vertical stacks of contained content in dynamic applications that can adjust to varying windows size and screen.

4.3 Windows communication foundation data services

As Silverlight is a client side application. Most of the time it interacts with database to access, save, and edit data. Many client side Silverlight applications can access the same database over the internet according to the roles given to them and WCF is an integral part of this process. The main point in this process is that after the database is created, the database sits on the user's machine running on the browser. Three main technologies are available to access this data via web service. The first one is a simple web service. The second one is WCF service that has easy means to provide

security. The third one is JSON (JavaScript Object Notation) that serializes the service. All these methods serve the purpose however for each operations like edit, create and delete data, a method must be created. In a large application this may turn out to be a problem. WCF data services come up with a different approach that resolves this problem. It uses REST (Representational State Transfer) that defines an address to the service and passing that service query. Whenever a query is made the service can figure out what it is required. [19,249-253]

In order to access data by WCF services there needs to be data source. In this theory part I will use two tables from the database I made for LiveMed electronic pharmacy. The tables are *city* for entering cities and *reservation* for sorting the reservations made. After the database is established, ADO.NET entity framework classes to the application are added to help develop it against the representation of a data source not the actual data source. In figure 17 when the tables are observed they may appear just database tables. Nevertheless, the relationship types are also specified. In the bottom portion of the figure where the mapping details are revealed, the default settings of the wizard can be changed, change the fields that map to each other, and field names can also be modified from there.

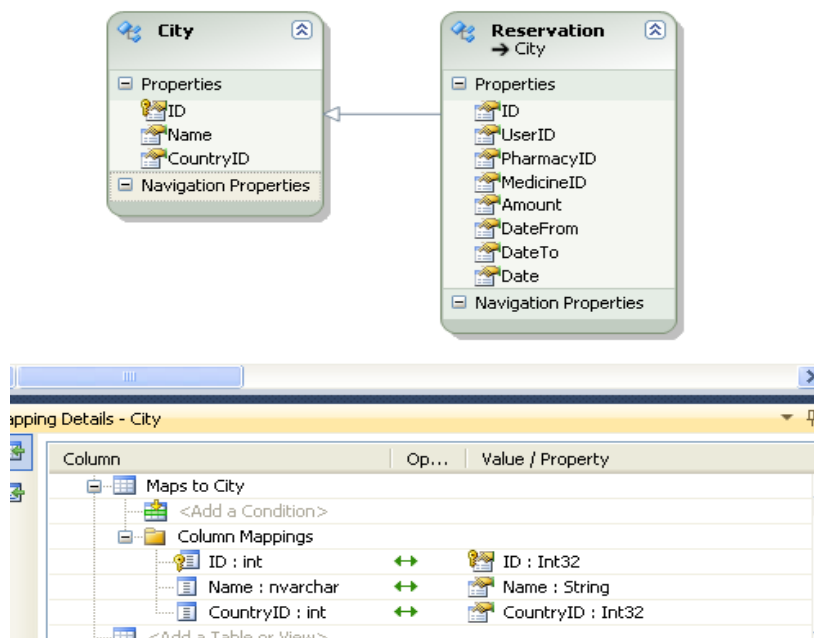


Figure 17 ADO.NET entity framework classes

The next step then is to add WCF data service by right clicking on the web project and choose new items. The wizard for creating is available. A `.svc` extension is provided for the service name which implies the fact that WCF data service is type of WCF service holding the same security, reliability and configurability properties. After creating the service on the wizard, it needs to be manually edited to include the class that contains all the entities the application needs to utilize. Once the setup is completed to access the database via the WCF data service, the Silverlight client application needs to reference the service to interact with the database.

5 Implementation of LiveMed electronic pharmacy

The LiveMed Electronic pharmacy is a Silverlight client application that uses WCF to communicate with database. I used Visual Studio 2010, Sql Server 2008, Silverlight SDK and Microsoft Expression blend tools to develop the application. When the application was up and running, it was hosted on IIS local server.

5.1 Requirement

The initial requirement for this project was to build an electronic pharmacy application where the user can make a reservation. The client side of the application was to be build with WPF (windows presentation foundation) and the host side to have a single pharmacy inventory database. The client and the host application were to use a web service to interact with each other. In a nut shell, the demo application was required to have one pharmacy database and one client page, which the two interact with WCF (web communication foundation). As the development of the application proceeded, the requirements grew. The client side was not expected to be limited being a desktop application. The business logic of the project required me to make it a web browser application. User authentication and administrative role assigning were the other two requirements that came into the picture after the development process started. The last requirement was to integrate Bing map with the system so that the locations of pharmacies using the system are easily traceable.

5.2 Use cases

There are two roles given for users while authenticating, the administrative and the normal user (Doctor) roles. Use cases for administrative role are as follows;

- Logs in to the system with user name and password
- Add, edit and delete country list
- Add, edit and delete city list
- Add, edit and delete pharmacy list
- Add, edit and delete medicine list
- Make reservation by selecting the desired fields
- Receives reservation receipt by email

This is one direction where administrative user follows. The user can also edit registered users and give or deny access to make reservation in the system. He/She has also access to the pharmacy pages to go through each pharmacy and see the locations aided by Bing map.

Use cases for non administrative doctor role are as follows;

- Registers to the system by filling detailed information
- Logs in to the system
- Goes to reservation page
- Chooses from available countries list
- Chooses from available cities list from the selected country
- Chooses the desired pharmacy from the city
- Chooses medicine from the ones listed under the pharmacy
- Selects the amount of the order
- Clicks on reserve button to make reservation
- Receives reservation receipt via email
- Goes to pharmacy pages and pages and browse through the desired ones to see address and location supported by Bing map

The non administrative user has also a master page option. However the information that can be set is only limited to setting user account settings like password, email, address and telephone.

5.3 Database

The database was made using SQL server 2008 tool. Figure 18 shows a data class diagram the visual studio generates while using the database in the application as entity framework.

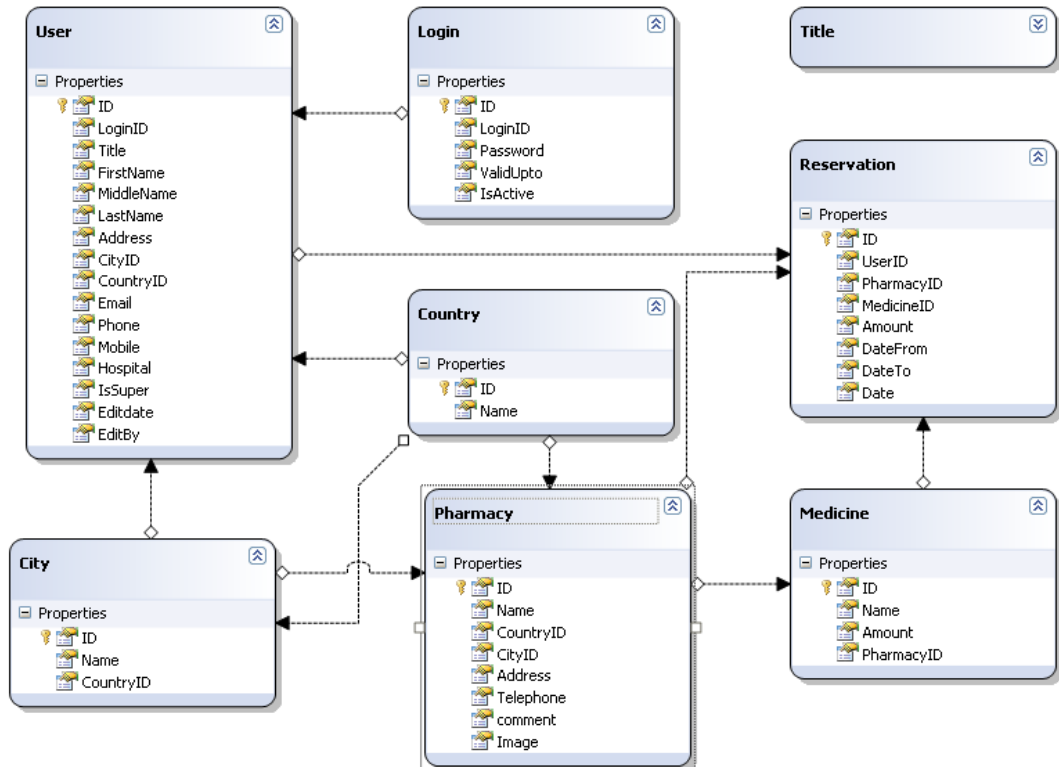


Figure 18. Data class diagram

As it can be seen in the data class diagram, the database consists of 8 tables. These tables with their specific property are as follows;

Country table: holds CountryId and Name properties.

City table: properties inside this table are CityId, Name and CountryID as a foreign key. It has a one to one association with country table which implies that the child table (city) is dependent on the country table. If there any data for the city table that has CountryID, then that specific CountryID should also be included in the country table.

User table: properties inside this table are UserId, LoginID, FirstName, LastName, Email, MiddleName, Address, Phone, Mobile, Hospital, IsSuper, EditDate, EditBy, CityID and CountryID. Most of the properties mentioned are standard properties used in many applications. The IsSuper property implies whether the user has administrative role or not. Whenever the user's information is edited the EditBy and EditDate properties help track the update details. This table is the child table for LogIn table, City table and Country table.

LogIn table: properties inside this table are LoginId, Password, ValidUpto and IsActive. The ValidUpto property specifies the period of time user is valid in the system. This IsActive property defines whether the user is active at the current time.

Pharmacy table: properties inside this table are PharmacyId, Address, Name, Telephone, Comment, Image, CityId and CountryId. It is the child table of city table and Country table.

Medicine table: properties inside this table are MedicineID, Amount, Name and PharmacyId. As it can be observed from the properties, it has a one to one relationship with the Pharmacy table. For each MedicineId there is a correspondent PharmacyId.

Reservation table: the properties inside are ReservationId, UserId, PharmacyId, MedicineId, Date, DateFrom, DateTo and Amount. It is a child table for Pharmacy, Medicine and User tables.

Title table: It holds properties of TitleId and name.

5.4 Development steps

Since I was running the whole application locally, I created a solution that has the client and host applications together. On the host side before I started creating the service I created seven classes under the data access layer. These classes contain methods that help business layer connect with the data and perform the required actions. The classes are CityDAL.cs, CountryDAL.cs, UserDAL.cs, PharmacyDAL.cs, UserDAL.cs, MedicineDAL.cs, LoginDAL.cs. In each class I created separate methods for checking,

saving, deleting, inserting and updating of data. The example below in figure 19 shows two methods used in CityDAL. The first method gets list of cities of a country and the second method saves new city to the data also assigning it the new city to the country that it resides under.

```

public List<City> GetCityList(int countryId)
{
    DataClassesDataContext dataContext = new DataClassesDataContext();
    return dataContext.Cities.Where(o => o.CountryID == countryId).ToList();
}
public int SaveCity(City city)
{
    if (!IsDuplicate(city))
    {
        DataClassesDataContext dataContext = new DataClassesDataContext();
        if (city.ID == 0)
        {
            dataContext.Cities.InsertOnSubmit(city);
            dataContext.SubmitChanges();
            return city.ID;
        }
        else
        {
            City uCity = dataContext.Cities.FirstOrDefault(o => o.ID == city.ID);
            if (uCity != null)
            {
                uCity.Name = city.Name;
                uCity.CountryID = city.CountryID;
                dataContext.SubmitChanges();
                return 1;
            }
            return 0;
        }
    }
}

```

Figure 19. Methods for data reading and saving

Then I created the WCF service. Visual studio creates an interface and a class. On the interface, the methods have operation contract as a function. The example below in figure 20 shows the operation contracts defined for updating city data.

```

#region City
    [OperationContract]
    List<City> GetCityList(int countryId);
    [OperationContract]
    int SaveCity(City city);
    [OperationContract]
    int DeleteCity(int cityId);
#endregion

```

Figure 20. Methods in WCF interface for updating data

The class created which has svc extension inherits the interface. Once the methods are defined in the interface they are implemented in this class. In figure 21 implementation of the operation contract defined in the interface is demonstrated.

```
#region City

    public List<City> GetCityList(int countryId)
    {
        CityDAL cityDAL = new CityDAL();

        return cityDAL.GetCityList(countryId);
    }
    public int SaveCity(City city)
    {
        CityDAL cityDAL = new CityDAL();
        return cityDAL.SaveCity(city);

    }

    public int DeleteCity(int cityId)
    {
        CityDAL cityDAL = new CityDAL();
        return cityDAL.DeleteCity(cityId);
    }

#endregion
```

Figure 21. Methods used in service class

Once the service is designed, the next step was to build the client application that uses the service. The first thing i did on the client application was to add a service reference. Then I built the design the Microsoft expression blend tool. I made 23 interfaces for each component.

Figure 21 shows the view folder holding all the user interface designs.

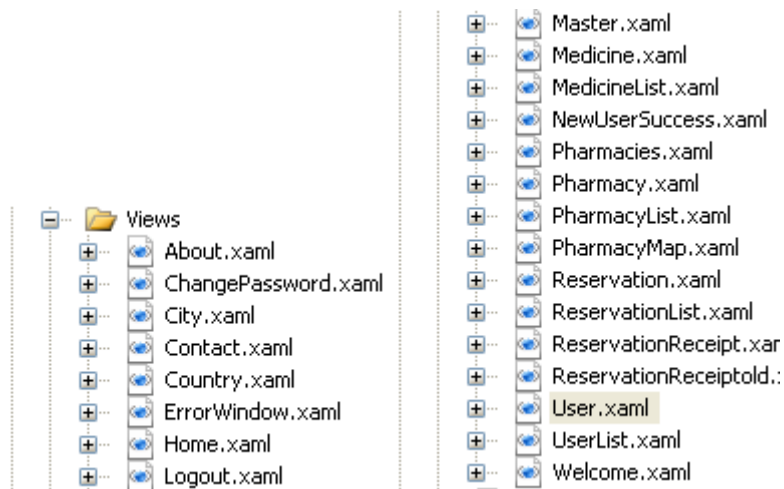


Figure 21 User interface component files

The last development process of this study was to publish the whole application in IIS (Internet Information Systems) local host.

5.5 Authentication and registration

There are two types of users to be authenticated in the system. These are users who have administrative role and users who can only make reservation and modify self settings. The login class and user settings are made in the host application and user settings are saved in the database. However I made a different class on the client side that uses isolated storage which is very effective in catching data.

Example below in figure 21 illustrates the methods used in authenticating the user.

```

public Authenticate()
{
}
public static string RedirectURL
{
    get
    {
        return "/Home";
    }
}
public static bool IsAuthenticatedUser
{
    get
    {
        IsolatedStorageSettings userSettings =
        IsolatedStorageSettings.ApplicationSettings;
        if (!userSettings.Contains("User"))
        {
            return false;
        }

        return true;
    }
}
}

```

Figure 21. Authentication class

If authentication fails to invoke, the system sends an error message and recommends user to register. Because of the business logic of the application, registration has multiple fields to be filled and important fields have asterisk beside them. All the fields to be filled have correspondent value on the database tables and are fetched to the database right the way.

5.6 Data handling

As explained above briefly, the seven classes were made to enable inserting, editing, updating, and deleting data. Only the user who has an administrative role can have access to the master page that holds these functionalities. Users with non-administrative role can only change their own data such as changing password, address, email and telephone.

Figure 22 shows an interface where user is able to add new pharmacy page to the system.

The screenshot shows the 'Pharmacy' administrative page. At the top, there is a blue navigation bar with the 'Live Med Pharmacy' logo and the text 'Logged in as mikias mulugeta'. Below this, the page title 'Pharmacy' is displayed. The form includes the following fields and controls:

- Country ***: A dropdown menu with the placeholder text 'Please select a country'.
- City ***: A dropdown menu with the placeholder text 'Please select a city'.
- Name ***: A single-line text input field.
- Address**: A large multi-line text area for entering the pharmacy's address.
- Telephone**: A single-line text input field.
- Comment**: A multi-line text area for additional information.
- Map**: A checkbox next to a small map icon.
- Image**: A checkbox for uploading a pharmacy image.

At the bottom of the form, there are three buttons: 'New', 'Save', and 'List'. An 'Upload' button is positioned to the right of the 'Image' checkbox.

Note : Fields marked with * are mandatory.

Figure 22. Pharmacy page with administrative view

5.7 Reservation

The reservation form starts by giving an option for user to select from the available countries. ComboBox is used for the user interface to show from the available list. The next option is to select from the available cities in that country. After that one can choose the pharmacy and the medicine with text box. It displays the dynamically selected object and the button corresponding to that opens a child window displaying the available pharmacies in the selected city or the available medicines in the database that correspond to the PharmacyId chosen in the previous step. Figure 23 below shows a method used for event handling of a button. The button functions to open a child

window that shows pharmacies residing under the chosen city. The first part of the method enables to handle errors in cases of user selecting the pharmacy button prior to selecting country and or city. The second method fetches pharmacy list in a row from database that correspond to the selected CityID.

```
private void btnSelectPharmacy_Click(object sender, RoutedEventArgs e)
{
    if (cmbCountry.SelectedIndex == 0)
    {
        tbError.Text = Message.SelectMsg("country");
    }
    else if (cmbCity.SelectedIndex == 0)
    {
        tbError.Text = Message.SelectMsg("city");
    }
    else
    {
        SR.City city = new SR.City();
        city = cmbCity.SelectedItem as SR.City;
        if (city != null)
        {
            PharmacyList pharmacyList = new PharmacyList(city.ID);
            pharmacyList.onRowSelect += new PharmacyList.SelectRowHandler(pharmacyList_onRowSelect);
            pharmacyList.Show();
        }
    }
}
```

Figure 23. Event handling method for a button

The user interface screen capture in figure 24 also shows what happens from user point of view when the button the methods of which described above is clicked.

| Pharmacy List | | | | |
|---------------|---------------------|---------|-------|---------------------------|
| | Pharmacy Name | Country | City | Address |
| ✘ | birhanu pharmacy | Finland | espoo | itätuulenkujä 8, espoo |
| ✘ | eskew pharmacy | Finland | espoo | kilonrinne 10d, espoo |
| ✘ | Espoon I Tapiolan a | Finland | espoo | Kauppamiehentie 6 , espoo |
| ✘ | Kivenlahden sivuapt | Finland | espoo | Merivirta 3, espoo |
| ✘ | mahi pharmacy | Finland | espoo | kilonkallio 10h, espoo |
| ✘ | mikias | Finland | espoo | siltakuja 2f, espoo |
| ✘ | Otaniemen sivuapte | Finland | espoo | Otakaari 11, espoo |

Figure 24. Child Window for viewing pharmacy list

After the country, city, pharmacy and medicine are chosen, the next step is to choose the amount of medicine and the interval of dates for pick up. In the reservation class there are methods used for getting data of the country, city and pharmacy fields, to save the reservation will all details including date and time, to save the reservation, to delete the reservation, to show reservation list and to send reservation receipt by email. All the values filled in making one reservation are fetched to be included in the body and head of the message.

The message image in figure 25 below shows the user interface of the message sent using the above method. On the email receipt it fetches message from the user's full name, time and date, the pharmacy name, the location of the pharmacy, the name of the medicine, the amount of the medicine and the preferred interval for pick up dates.



Figure 25. Reservation confirmation mail

5.8 Mapping implementation

The first step to implement Bing map in the application is to create a developer account and generate a key to be accessed by the application. Then the Bing Map Silverlight control that has the Bing map Silverlight control assemblies and the offline Bing maps Silverlight SDK Were downloaded and installed. To include the service in the application, Microsoft.Maps.MapControl.Common.dll and Microsoft.Maps.MapControl.dll were referenced to the client application from the directory it was saved while installing the system. Bing Map takes only latitude and longitude information. However, the application requires taking the address of the location and locating it on the map. To en-

able that, I added the GeocodeServices Bing Map web service on the client application. In the application the address of the pharmacy is recorded by the user. When any user browses the pharmacy list page, all the details each pharmacies are presented including the map of that pharmacy. A button assigned for this purpose loads the map on a child window which uses the pharmacyMap view partial class. In these class methods for initializing the Bing Map web service, getting the map key, setting the location, showing the results and handling wrong addresses are used. The code below in figure 26 shows a method used to set credentials using a valid Bing Map key and to set the full address query.

```
private void SetLocation(string address)
{
    GeocodeRequest geocodeRequest = new GeocodeRequest();

    geocodeRequest.Credentials = new Microsoft.Maps.MapControl.Credentials();
    geocodeRequest.Credentials.ApplicationId = mapKey;

    geocodeRequest.Query = address;
    GeocodeServiceClient geocodeService = new GeocodeServiceClient("BasicHttpBinding_IGeocodeService");
    geocodeService.GeocodeCompleted += new EventHandler<GeocodeCompletedEventArgs>(geocodeService_GeocodeCompleted);
    geocodeService.GeocodeAsync(geocodeRequest);
}
```

Figure 26. Method used in Bing Map

The map figure below in figure 27 shows an address assigned to Bulevardi 31, Helsinki, Finland. It is zoomed in and out further for better view and it can be seen on the right corner of the picture it can be shifted to different from road view to aerial and labels views.

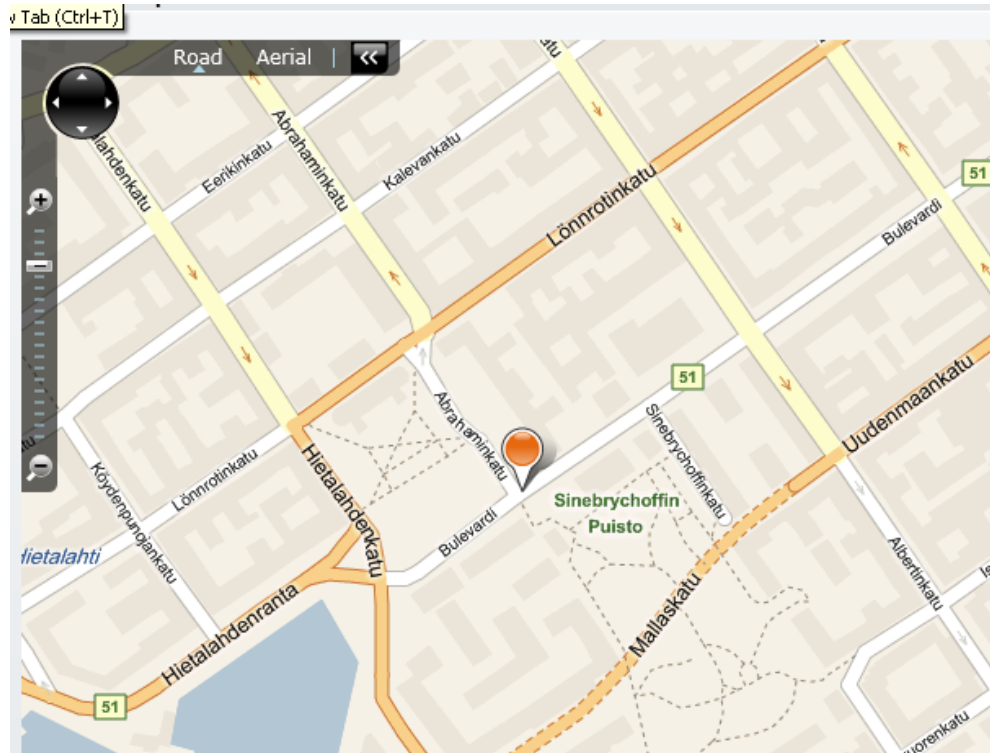


Figure 27. Road view in Bing map

5.9 Publishing

Before publishing the application the ServiceReference.clientConfig file was edited in order to change the client's end point address to the real one. The code in figure 28 is the portion of the Clientconfig file where the end point address is specified.

```
<client>
  <endpoint address="http://localhost:1430/LiveMed.svc" binding="basicHttpBinding"
    bindingConfiguration="BasicHttpBinding_ILiveMed" contract="ServiceReference.ILiveMed"
    name="BasicHttpBinding_ILiveMed" />
</client>
```

Figure 28. End point address configuration

In the host side a database connection string and application setting for email sending and Bing map key were set. Those settings are found in the WebConfig file.

Figure 29 shows the Windows IIS (Internet Information Systems wizard) manager- where the application was hosted after being published by the visual studio settings.

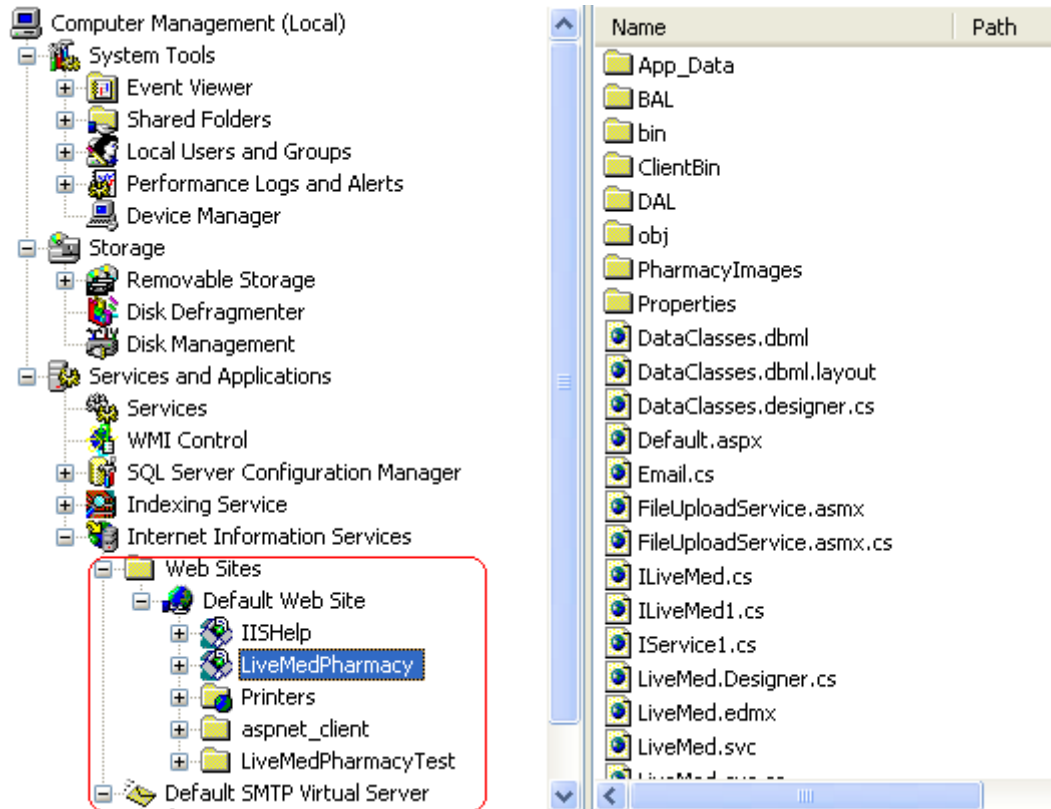


Figure 29. IIS manager

The publish web site button opens a wizard that holds information for publishing the application. First the publish method needs to be specified. The options for this are web deploy, file system, FTP and FPSE. For this study the file system was chosen. If the option web deploy is to be chosen, the service URL needs to be specified. I made sure that the IIS environment is set properly before adding the file system to it. In the IIS manager I created a virtual directory and in the Alias input name, the physical directory of the published file system was placed. The last step was to convert the virtual directory to application by setting the convert to application option. The web site now is published in the IIS and ready to be launched.

6 Conclusion

Today, almost every company possesses a database. In this information technology age, the need to have a medium to share these data that reside in different servers is imperative. Service oriented applications come up with the right business logic and implementations to make this happen. The aim of this project was based on this concept. It uses remotely hosted inventory database to solve a business logic of a client application.

The purpose of the study was to build a Silverlight client application that is connected with multiple inventory databases for manipulation of data according to the required purpose. I chose the web communication foundation web service framework to program the web service. By the time the project was completed, it had a reservation system of medicines that accesses data remotely from a database.

When building a web service there are many steps involved. The main ones are setting the ABCs of the service. These are the address of the service where it is available to be used by the client, the binding types to be used based on the type of connection the web service is intended to provide and finally, the contracts referring to the set operations that specify what the end points communicate with the outside world. Once services are constricted, the hosting environment becomes an issue. Each hosting environment has its own set up and security issues to deal with. Building a web service and not having a client application to use it makes the whole service oriented application incomplete. As any web application, the client application needs high consideration in designing the user interface. While implementing the Electronic pharmacy client application, I found Silverlight very handy for implementation of server side functionality with rich design.

To achieve the primary goal of the project, a demo database that holds data on different pharmacies and medicines available for each pharmacy were made. Windows communication foundation framework was used to design the web service and make the database available for connection. In the data access layer of the application I built classes that enable the user to add, insert, view and delete data. A client application that uses web service to connect with the host application was also made. In achieving the secondary goals of the project, I added different user roles that have different de-

degrees of access in the application. For the administrative role, extra functionalities such as adding data control access for users, pharmacies and medicines. I also added Bing map functionality to the application in order to enable users to view the precise map of the pharmacy location where they want buy their medicine.

The future of the Electronic pharmacy application should be having a mobile client application and make the reservation system in such a manner that a GPS system is used to automatically designate the registered pharmacies closest to the user and present the search results based on locations.

References

1. Papazoglou M.P. Web services: principles and technology. England: Pearson Education Limited; 2008.
2. Andrew T, Curbera F, Dholakia H. Business process execution language for web services. BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems; 2003.
3. Seng P. Web services programming tips and tricks [Online].
URL:http://www.ibm.com/developerworks/websphere/library/techarticles/0511_phung-lu/0511_phung-lu.html. Accessed 12 April 2011.
4. W3Schools. SOAP introduction [online].
URL: http://www.w3schools.com/SOAP/soap_intro.asp. Accessed 12 April 2011.
5. Sheryl M, Rick S, Michael W. Oracle database Java developer's guide. Oracle; 2004.
6. Kuassi M. Web services enable your database [Online]. URL: <http://www.java.sys-con.com/node/39726>. Accessed 14 April 2011.
7. Windows Communication Foundation [Online].
URL: <http://msdn.microsoft.com/en-us/library/dd456779.aspx>. Accessed 14 April 2001.
8. Anuj V. WCF versus ASMX services[Online].
URL: <http://www.anujvarma.com/BlogEngine.net/post/2010/08/02/WCF-versus-ASMX-services.aspx>. Accessed 15 April 2011.
9. Ian G, Matthew A, Jesse Liberty. Building windows, Web and RIA applications for .NET for c# 4.0. USA: O'Reilly Media; 2010.

10. Istavan N, Andras V, Adam G. Visual Studio 2010 and .Net 4 six-in-one. Indianapolis, Indiana: Wiley publishing; 2010.
11. Andrew T. Pro C# 2008 and the .NET 3.5 platform. USA: Springer-Verlag; 2007.
12. Pablo C, Curt K, Fabio C. Windows communication foundation With :NET 4. Indianapolis, Indiana: Wiley Publishing; 2010.
13. Michele L. Hosting WCF services [Online]. URL: <http://www.devx.com/codemag/Article/33655>. Accessed 20 April 2011.
14. Chris P, Dennis M. Pro WCF practical SOA implementation. New York, USA: Springer-Verlag; 2007
15. Bing Maps [online].
URL: <http://msdn.microsoft.com/en-us/library/dd877180.aspx>
16. Silverlight [online].
URL: <http://www.silverlight.net>. Accessed 23 April 2011.
17. Laurence M. Microsoft Silverlight 4 step by step. USA: O'Reilly Media; 2010.
18. Laurence M. Silverlight 4 unleashed. Indianapolis, Indiana: Pearson Education; 2011.
19. Mahesh K, Philip P. Silverlight 4 for dummies. Indianapolis, Indiana: Wiley Publishing; 2010.

Title of the Appendix

Content of the appendix is placed here.