Avdelningen för datavetenskap vid Institutionen för Informatik och Matematik

EXAMINATIONPROJECT 2003:D34

WebADmin - A faster way to administrate Active Directory

Anders Nilsson

HÖGSKOLAN
TROLLHÄTTAN • UDDEVALLA

Examination Project

Högskolan Trollhättan · Uddevalla
Institutionen för Informatik och Matematik

Essay for philosophy candidate examination in Computer science

WebADmin
A faster way to administrate Active Directory

Anders Nilsson

Examinator:
Doctor Samantha Jenkins Institution of Mathematic and Informatics

Handledare:
Peter Johansson EDS Sweden AB

Trollhättan, 2003

2003:D34

DEGREE PROJECT

WebADmin

A faster way to administrate Active Directory

Anders Nilsson

Abstract

Is there a way to make Active Directory administration faster and easier? The purpose of this project is to find out if there is. I will focus on the most common task of AD administration, group memberships, and try to create an application with which users can edit these memberships without having to install Microsoft's own tools.

The product became an application that is presented to the user in a web-browser. It makes it possible to connect to any specified server in the directory and make the changes, or to the fastest responding one. It has also the possibility to work in a certain part of the directory-structure, to make it find objects faster. Further on, it is also possible to add more then one object to a group in a much easier way then in Microsoft's tool, as well as adding one object to several groups at once, or a lot of objects to a lot of groups at once.

The result of the product is that it takes less time to do the same things as before, as well as it makes it possible to do it from any computer without having to install any software. This saves time and money for the company that could be spent on other things.

Publisher:	University of Trollhättan · Uddevalla, Department of Informatics and Mathematics Box 957, S-461 29 Trollhättan, SWEDEN Phone: + 46 520 47 50 00 Fax: + 46 520 47 50 99		
Examiner:	Doctor Samantha Jenkins		
Advisor:	Peter Johansson, EDS Sweden AB		
Subject:	Computer science	**Language**:	Swedish
Number:	2003:E999	**Date**:	November 4, 2003
Keywords	Active Directory, administration, groups, web-application		

Index

Bilagor

Nomenclature

ActiveX – A technology used to make software applications available on the web.

Active Directory Object – An object that exists in the Active Directory database. It contains information about a specific object that exists in real life and is connected to the network. In this essay an object represents a user or a computer in Active Directory

AD - Active Directory. A directory service used by Windows 2000. In this database, Windows keeps its information about users, computer, groups and all other objects that is part of the domain.

ASP+ - ASP Plus. This is another, shorter name for ASP.NET, which is the sequel to ASP 3.0.

Domain - A term used to describe a collection of computers, servers, users and services that are available in a separate part of a network. More then one domain can be connected to each other to make the services in one domain available to the users in another.

Domain Controller (DC) – A server that holds the Active Directory on the network and handles all queries against the AD. A domain can contain several DC's, spread throughout the company, to make faster response times. This since the response times are depending on the network bandwidth between the clients and the servers.

DLL – Dynamic Link Library. A file that consists of compiled code. This is usually a set of methods that is made available to more then one application by placing them in a separate file.

File-share – A way of sharing files and applications on a server, making it accessible to other computers on a network. It can be public and protected, so one have to enter a username and a password t access it.

IE5.5 - Internet Explorer 5.5, which is a web-browser, and the standard one in Windows 2000

MFC – Microsoft Foundation Classes. This is a framework of classes that Microsoft has created for developers to make their coding easier. It consists of the code for all the common parts that is used when creating Windows applications, such as code for button textboxes, manipulation of text and many more. It is available in both Visual Basic and C++.

MMC - Microsoft Management Console. This is a application used to perform a large number of administrative tasks. Each service that needs to be administrated has its on plug-in that is run in MMC, and through this plug-in it can be reached, even though the service may not be running on the same computer as MMC. This means that you can use MMC on a client computer to administer a service that runs on a server.

OU – Organizational Unit. A object in Active Directory that is used to organize it into multiple levels, to make it look like the company does in real life. If the company is spread over several countries, each country can be represented by its own OU, witch contains all the objects in that country's offices.

Win2K - Windows 2000. This is one of the most common operating systems in PC's. There are also a range of server variants available as well.

Win32-application – An application that runs under any of the Win95, Win98, Win ME, Win2k or Win XP operating system.

1 Introduction

The idea for this project came to me during the autumn of 2002 when I still was employed by EDS here in Trollhättan. During the year we had replaced somewhere around 2000 computers at SAAB and reinstalled an additional 2500 computers. All of them from Windows NT4 or 95 to Windows 2000. Alongside with the installations came other administrative tasks to keep the computers running. The main tool for this is Microsoft's Management Console. Since this tool has to be installed on the machine it is to be used on, it is rather clumsy to use out in the field at the clients computer. The idea came to me that this could be accomplished in a easier way, and most important, we should be able to do it at the clients computer without having to install any additional software. The problem was that there was no time or resources available at this time to accomplish this.

As my employment ended at the beginning of 2003, I got the opportunity to do this project as my examination project at school. I contacted my former boss and confronted him with the idea. He bought it at once, and two days later I started it.

I would like to thank the people at the TFS department at EDS here in Trollhättan for their support and ideas along the project, as well as without any hesitation making a computer and a desk available to me right away.

A special thanks to Peter Johansson for technical support and Ulf Johansson for testing every possible use case in the application and finding the bugs so I could fix them right away.

1.1 Background

During the second half of 2001 and for the whole 2002, I worked at EDS Sweden AB in Trollhättan. One of my tasks here was to install and deliver new computers to the customers at SAAB. We were also responsible for the installation of the applications on the computers. This is mainly done by adding the user to a certain group in AD, and when the user logs on, the server automatically installs the software that corresponds to that group on the clients. But sometimes, it's forgotten to add the user to a group, or there is a miss in the order, and the user should have more programs then what was ordered. To solve this, the person that delivers the computer to the user has to return to his desk in the office. There he has to start MMC and add the user to the right group in AD, before he can return to the user to make sure it all works as it's supposed to. This causes a lot of unnecessary waste of time before it can be fixed and the user is satisfied. This is time-consuming, as it can take up to 20 minutes to go from the office to the user. To this, add 10 minutes for the work at the office before returning to the user. So it can result in 50 minutes loss on a thing that could be solved in 3 minutes if it was possible to use MMC at the users' desk. Unfortunately, this is not possible due to security

restrictions on the client pc. This gives a less satisfied user and it makes EDS appear as a clumsy and lazy company to he user, which of course is bad.

As I am no longer employed by EDS, and is about to do my examination project here at HTU, I took this idea and presented it to my former boss at EDS, and offered myself to do it for them as my examination project. This was fine with them, as they don't have the time to do this kind of tool themselves at the time being. Since I have worked as a technician myself, and came up with the idea at the time, I have a pretty good idea of how it should be done. But I will also consult the current technicians if they have any requests or ideas that could be implemented in the solution.

1.2 Purpose and goal

The purpose of this project is to create a tool that eliminates the process of going back to the office to perform a certain action, and the moment later, return to the user to verify that it works. I am going to try to make a tool that can access AD from the users computer. It must be possible to use it without logging out the user from the computer. If I can accomplish that, the extra time spent on each case if software is missing will be minimized to a few minutes instead of, as before, up to 50 minutes. This will make the individual user happier as he can continue working with his new computer in less time. It will also make the customer, in this case SAAB, happier as more cases can be solved in less time. I will also try to make the application faster to work with then Microsoft's MMC-plug-in.

2 Prerequisites

There are a number of prerequisites that narrows the number of possible ways to solve this. Some of them is on the server-side, and some on the client-side. Active Directory is also a prerequisite that need further explanation.

2.1 Active Directory

Probably the most important prerequisite in this project. If we break it down to the basics, it is a database that keeps information about all the objects, both real-world, as computers, and abstract ones, as user groups and security policies, that are part of the domain. For more information about Active Directory, se appendix A.

2.2 Server-side

The servers are based on Windows 2000 Server. They use Active Directory to manage all users, computers and other objects that form the domain. Internet Information Server is the web-server that is used.

2.3 Client-side

The clients are running Windows 2000 Professional as operative system. As standard, they come with Office 2000 and Internet Explorer 5.5. A number of security policies are applied to each computer that logs on to the domain. These are restrictions in what the user can do to the computer and what kind of content that can be viewed or downloaded from the internet or intranet. Default is that the individual user can't install any software on the computer by himself, and only has access to the intranet.

2.4 Limitations

Since I brought the idea to the company, and not the other way around, EDS hasn't set any real limitations on the project. Instead they rely on my experience since I was employed by them to make the tool appropriate for the situation.

I on the other hand have set a few limitations, to prevent the project from growing to an uncontrollable mass that will take way too long time to finish inside the frames of this course. The tool will only handle group memberships, i.e. make it possible to add users and computers to a group in AD and remove them from groups. Another limitation is regarding how the tool is available to the user. It should be available for use without any need of installation or downloading of components to the client. This rule out certain technologies that could have been used, as ActiveX.

3 Method

How do I create such a tool? The first thing to consider is how it should be available to the user. To do this I have to look at the environment in which it is going to be used. Another thing I have to consider is how much changes that must be done to the environment to make the application work.

3.1 Available Methods

3.1.1 Win32-application

Using this method, I can create a standard Windows application that can be placed on a fileserver somewhere on the network, so it is available to the clients anywhere on the network. If I choose to make it as an win32-application, I have to create the application in one piece. This because there can be no needs of registration of dll's on the client, as I am supposed to still be logged in as the end-user on the machine, and they don't have the rights to do that. This causes no major problem. The only effect is that the application may become a little messy, from a code point-of-view, as all the code has to be in one single file. Further, the user has to make two logons to connect to the AD. One logon to access the fileserver and start the program, and another in the application. The later one is used to supply the user credentials that is used when connecting to the AD.

It may occur as a disturbing thing that one has to logon more then one time with the same username and password. But it is fully possible to create the tool using this method. With this method, the following languages are available:

- C++
- Visual Basic
- Visual Basic.Net
- C#
- Java
- J++

3.1.2 Web-application

Using this method, I can create an interface that the user can reach using a web-browser. This gives a single login when the user accesses the application on the web server. The information from that login can then be used in the application when connecting to the AD, if the web server and its authentication-process are set up the right way. If I decide to use this method, the following techniques are available for me to write it in.

- ASP
- ASP.NET
- PHP
- JAVA
- ActiveX

3.2 Choosing a method

So, which method is the best one to use in this case, and what technique should I use to implement it? Since the primary goal is to save time for the users, this will be my primary consideration when choosing method. Along with that I have to make sure the application causes as little impact to the existing systems as possible. And last, I want to create a nice and clean application, from a code point-of-view.

How do I measure the speed of an application before it's created? I could of course use some kind of simulation to estimate how long time it takes to access the application, how fast the user handles it, and how fast it communicates. All this depending on which method I use. But that would take to much time and the level of effort needed is more then this project requires. So I have to decide it using simple logic instead.

If I create a win32-application and place it on a fileserver I need to place it on a file-share that is protected. This to prevent unauthorized access of the application. Thus it takes one logon to access the application. Once I have started the application I have to

supply the username and password again to the application, so it can use them to connect to the AD to get the permissions to change group memberships. Already we have performed four events just to get connected to the AD. First enter the path to the file-share, then authenticate against the file-share. After that the application has to be started, and then we have to enter the user information once again. What about a web-application? Can I use that method to eliminate any of the steps? Let's see. First I have to access the application on the web-server. This is done by entering a URL in the web-browser. To be able to load it I have to authenticate myself against the web-server. When this is done, my user credentials are stored on the web server as long as I have the browser open and pointed to that address. This way, the username and password is available to the application without me entering them one more time, and yet the application is started and connected to the AD in just three steps. This gives the web-application method an advantage already before the user has begun using it. But can I create an user interface on the web that is just as easy and fast to use as I can in a win32-application? Well, with the old traditional web-pages I couldn't. With today's technologies, such as ASP and ASP.NET there is no problems of achieving this, as there exists graphical controls as buttons and list boxes etc for web-pages that looks just like and functions just like the controls used in win32-applications.

Therefore, I am going to create a web-application.

The next choice is which technology I'm going to use. See appendix B for information on which techniques that are available and further information on how they work. Based on that information, on the prerequisites, and that it should cause as little interference with the existing systems as possible and that I want to make the code look good and easy to read and modify if needed, I have decided to use ASP.NET as the technology in which to create the application. After that, I have to decide in what language I'm going to write it in. The available ones are:

- Visual Basic.NET
- C#
- J++
- C++

Since this has no real effect on the performance of the application, I will base the decision upon my personal preferences. As I have used all languages but C# before, I feel that it could be interesting to write it in C#, so I can learn it and see if it is a good programming language or not.

So, the application will be made as a web-application in ASP.NET using C# as language. Since C# is new to me, I will need some technical references to it, and I found that the Microsoft Developer Network [1] contained all that was needed for the task.

4 Design

Now that it is stated how the application is going to be created, it has to be decided what should be possible to do with the application. It has already been stated that the need is to be able to edit group memberships in Active Directory. Well, that's kind of a vague description. I have to specify how to edit the group membership so i can create a user interface that meets the needs in the most effective way. Another design issue is how the code is created and stored. This to prevent from writing code that is used more then one time.

4.1 Application Functionality

As said before, a way to change group memberships in AD is needed. There are three ways of doing this. One can add one object (user or computer) to one or many groups at once. Or one can add one or many objects at once to one group. And last, there is the need to add several objects to several groups at once.

And all of this, if possible, should be made in shorter time then in Microsoft's tool. One of the main reasons to why the MMC snap-in can be experienced as kind of slow, is that you work with the whole directory all the time. The structure of the directory is represented as a tree-view to the left. When you select an OU or an object in the tree-view, all its children is fetched from the AD and displayed in the list-view to the right. If the OU contains many objects, this operation can take quite long time. Therefore the solution to this project will handle only as many objects as needed. This will be accomplished by letting the user search for the objects needed, instead of having to browse for it, like in the MMC. Another disadvantage of the MMC plug-in is that it can only handle one object or group at time, creating extra work if you need to add many objects to many groups. This will also be made easier by this project.

During the development EDS requested two functions that weren't planned from the beginning. One was that they wanted to be able to save all members of a group to a text file.

The other function that was requested during the work was that it should be possible to select in which OU one will work, and against which server the application performs the queries. This should also be saved from time to time. When the application starts, all queries will be made against the fastest responding server, until the user selects a server, or a preferred server is read from the user settings file. And all queries will be made against the root of the AD until the user selects a different OU or the users preferred one is read from file.

4.1.1 Edit Group

The first page will be called "Edit Group" where several objects can be added to and removed from one group. It must be possible to search the AD for groups to edit, then to

select one of the search results and display information about it, such as name and which objects that are members of this group. It must also be possible to search for objects that will be added to the group.

4.1.2 Edit User

The second page will hold the "Edit User"-part. It is almost the same as the Edit Group-page, with the difference that one starts searching for users or computers instead of groups, and show the information about them. And one must be able to search for the groups that the objects are to be added to.

4.1.3 Add Many

The third page is intended to use to add several objects to more then one group at once. Therefore, we will need two different search-functions available at the same time. One to get the objects and one to get the groups. Then the user just has to select the objects and the groups and press a button, and the selected objects will be added to the selected groups.

4.2 User Interface

How can this functionality be achieved? The easiest way to do it, and the least complicated for the user, is to make three separate web-pages. One for each way of adding objects. These three pages should be accessible through a menu, where the users can choose which part of the application he wants to use. The most common way of doing this in a web-page is to divide it into two vertical frames. One narrow frame to the left that displays the menu, and one larger to the right where the chosen part of the application will be displayed. To add more functionality there will be a banner-frame at the top that cuts across both the vertical frames, where various information can be shown.

Each of the graphical controls described below, such as textboxes and list boxes will have explaining labels as needed.

This section will only describe the graphical layout of the pages. Screenshots from the different pages and modes can be viewed in appendix C. The functionality behind the pages is described in appendix D with several UML-diagrams, and the code is found in appendix E.

4.2.1 Edit Group

When a person uses an application, or a web-page, it access the information and functions in it just as reading a book, left-to-right, top-to-bottom. Therefore the functions that is used first on the page should be placed at the top-left corner of the page, and then placed in the order that it is used, continuing to the left and then down, if

several rows is needed. Using this principle there will be two dropdown list boxes and two buttons on the top line of the page. One list box showing all the OU's in the AD and one showing all DC's. The button saves the current selection in the list boxes, one for the OU and one for the DC, into a file. The file should be named after the user that is using the application.

The next line will hold the two textboxes where the user enters which groups or objects that should be searched for. One of these will be invisible as long as the user searches for groups. Under these textboxes there will be a button that initializes the search. To the right of these, there will be a textbox that displays the name of the selected group. Under these textboxes two list boxes will be placed. One to the left that shows the result of the search and one on the right that shows the members of the currently selected group. Under these there will be five buttons, two under the left list box and three under the right one. The first to will add a object to a group and the second one will take you back to "group-search"-mode. The three buttons under the right list box will either take you to "object-search"-mode, remove a object from a group, and last, save the members to a text file, named after the name of the group.

The last section of the page will contain a large textbox that will show messages from the application., like how many users that is found, how many objects that is added to the group etc.. To the right of this there will be a button that clears this textbox.

4.2.2 Edit User

This page will look almost exactly as the "Edit Group"-page. The only difference is that the button that says "Add objects" on that page here says "Add groups". And the info that is shown to the right has one more textbox. The two textboxes shows the logon-name of the user, and the users real name. In the list box to the right, the groups that the user is member of, is listed.

4.2.3 Add Many

On this page, as well as the two others, will contain the objects needed for selecting OU and server to work with. The thing that differs this page from the other two is that there is no information displayed about the selected objects or groups. The next section of the page will be divided in to two parts, one to the left and one to the right. The left one will hold the search-fields for objects, and the right one will hold the same for groups. The left section will have two textboxes, a button and a list box. One textbox for users and one for computers. The button starts the search and the list box displays the results.

The right section will have one textbox where the user enters which groups to search for. Below that textbox there will be a button that starts the search for groups and a list box where the result will be displayed.

Once the searches is done, the user just has to select the desired objects in the left list box and the groups on the right one, and press the button to add the objects to the groups. When the object is added to a group, a message will be written in the log window.

4.2.4 The Menu

I have tried to keep everything as simple as possible, so it will take so little time as possible to load the application. And the menu will follow that example, and consists of simple text links. The first one takes you to the page that probably will be used the most at EDS, because of their implementation of software deployment. That page is the "Edit Group"-page. The second one will take you to the "Edit User"-page and the last one to the "Add many"-page.

4.2.5 The Banner

This window is used to display various information that could be useful to know for the user. It is also used to display the application logo, which is placed in the top-left corner of the page. At start, I will display try to show the hostname of the computer that you are accessing the application from, and the username of the currently logged in user on that computer.

4.3 Code Design

Another thing to consider when creating an application like this is to make it easy to adjust for the company, if they have to change anything in the AD that affects the application. Therefore there will be a separate class that holds the code hat communicate with the AD. This class has to be made independent of object-types and server-addresses and al other things that may change during time. Instead, all of the above three pages that edit the AD will implement this class and send the relevant information to the current instance of the class when needed. All basic information that may change over time is saved in a plain text file that the application reads every time it is started. As the application is now, there is only one post in this file, and that is the string that represents the LDAP-name of the domain. This because it should be easy to move the application between domains.

A new thing with ASP.NET is that it is possible to separate the code from the web-page, using a technique called "code-behind". For more in-depth information on this, see appendix C. In short, this gives me a possibility to write the code in a separate file from the web-page. And even better, I can write the code in a object-oriented way, instead of the in-line spagetti model that often is used in ASP 3.0 and older.

This gives three separate code-files for the three pages, each with its own web-page linked to it. This makes it a whole lot easier for further development and fine-tuning of the application. If a web designer wants to modify the design and layout of the

application, he just has to edit the web-page files, without risking editing the application code by accident. And a programmer can edit the code without knowledge about web design, if he has to make adjustments in the application in the future.

5 Result

When the application was finished I had it up and running on my test server where I had an Active Directory installed and designed as a miniature of the live AD that is used. There a technician, that is supposed to use the live version later on, tested all the functions. He found that it was working just as it was supposed to and that it was really useful. It saved quite a lot of time when compared to the standard Microsoft tool that is currently used.

Due do security restrictions, I can't install the application on the live AD that is used, so I had to create an installation package and deliver that to the company instead. After installing it, they only have to change the LDAP-string in the configuration-file to make it work.

6 Analyze of the result

So, did the final product meet the purpose and goal that was set at the start of the project? Yes, it did. The result is an application that is accessible through a web-browser. It can perform all the actions that it was intended to do, as well as the extra ones that EDS requested. It is faster to use then Microsoft's MMC-plug-in too, just as intended. The increase in speed came from the fact that the application has optimized the traffic with the AD, by just requesting the objects that is needed by the user for the moment. This results in less data that have to be sent from the AD to the client, which gives a faster working application. I also implemented the option to choose which Organizational Unit in the AD that the searches is made against. This gives a quicker response-time, as there is a smaller part of the AD that has to be searched for the objects requested. It also gives smaller search-results if you search with wildcards. Also, the operation of adding a user, or more, to a group is performed in less steps using WebADmin, then it requires in Microsoft's product.

The only thing I didn't manage to do was to install it on the live AD and use it there, due to security restrictions. Therefore I leave this part to the company too.

7 Conclusions

The intention of this project was to create a faster and easier way for the technicians, at EDS here in Trollhättan, to view and edit group memberships in Active Directory. As it is done today, one has to use Microsoft's Management Console Plug-In, which has to be installed on the computer you want to use it on. This gives a un-flexible tool to use.

With the result of this project, WebADmin, there is a faster and easier way to do all this. The application is available through the web-browser and doesn't require anything to be installed on the client. The conclusion is that it is possible to create a faster and easier application then the MMC Plug-In that is shipped with Windows. It is also possible to make it available to the clients without any need to install the software, which gives a very high availability. And through very simple adjustments, I made it possible to use not only in Trollhättan, but on every place in Europe where EDS may need it.

But this project doesn't make life all that wonderful. There is still much improvement that can be done. As it is now, it only gives the user a possibility to change group memberships. All other administrative tasks has to be made through the MMC Plug-In. With further development of WebADmin, the time spent on administrating the other parts of the Active Directory could be cut as well, and more money saved, as these tasks could be made faster and easier as well. And available on any computer connected to the network.

References

1 Microsoft Developer Network Library [Electronic], Microsoft Corporation. Available: < http://msdn.microsoft.com/library/> [2003-08-04]

Appendix A
Active Directory

What is Active Directory? Here I will try to give an explanation that is easy to understand, and that gives you all the information you need to understand the purpose of this project.

If you break it down to the basics, Active Directory is nothing more then a database that contains information about all objects that exists on the network. For a user object as example, it holds information about user name, real life name, address and whatever you like. One of the advantages of Active Directory is that you can define your own information-posts in an object, and then use them in applications. The three most important parts of the directory is user-, computer- and group-objects. The user-object holds information about the user, as said above. The computer holds the same information about the computers. The group-object is used to collect several objects into one group. This way, you can edit a property on many objects at once. Using this technique also makes it possible to apply policies to users and computers. This can range from security-policies that gives the user or computer rights to access certain resources on the network, as printers or file-shares. This objects, printers and file-shares are also parts of Active Directory.

Another feature of AD is that it is possible to deploy software to computers automatically. What you need is a software package that can install itself without any user intervention. Then you add the computer to a group in AD that applies this software to all members of the group. Next time the computer logs on to the server, the server checks if al the software that should be installed is installed. If it is not, the installation starts and the computer receive the software. This all happens automatically and the user doesn't have to do a thing.

If you understand this feature of the AD, then you should be able to understand the purpose of this project. Of course there are a lot more features and possibilities available in AD, but that is not necessary to bring up here. Fore more information I would recommend the Microsoft Developer Network Library, available at http://msdn.microsoft.com/library/

Appendix B
Web-application Programming Techniques

PHP

PHP is a script language that is very suited for development of web applications. It has native support for LDAP as well as use of COM-objects. This means that I could use any available technology that supports ADSI to create a COM-object which we implement through PHP. Though, in order to implement PHP, a PHP-engine has to be installed on the IIS. The reason for me to review this technology is that PHP is free of charge to use.

Java

In the world of Java, there is an API called JNDI available, which makes it possible for Java developers to connect to directory services in various ways and perform various administrative tasks, as well as store and retrieve Java objects in the directory. Another part of Java is Web-services, which can be used to create graphical user interfaces in Java. These two API's gives me a possibility to use Java as a tool to develop the application without the use of external COM-objects.

ActiveX Controls

ActiveX Controls is a technology used to present an application in C++ or Visual Basic to a user using Internet Explorer as host, which makes it possible to distribute an ActiveX Control through the web. Since it is written in C++ or Visual Basic, it is possible to implement ADSI into an ActiveX Control. Using this, I could create a Windows-like user interface for administering Active Directory through the web. With Windows-like I mean that I could make use of all objects and controls that are available when writing a standard Windows application.

Component Object Model (COM)

COM is a system for creating binary software that can be distributed and accessed through different platforms developed by Microsoft. In short, it could be said that a COM object is a set of classes written in C++ or Visual Basic, which has been compiled into a binary package that other applications can access. This method allows me to store all AD access methods in a binary on the server, and the only thing that the user can access is the user interface which uses the COM object.

Active Server Pages (ASP)

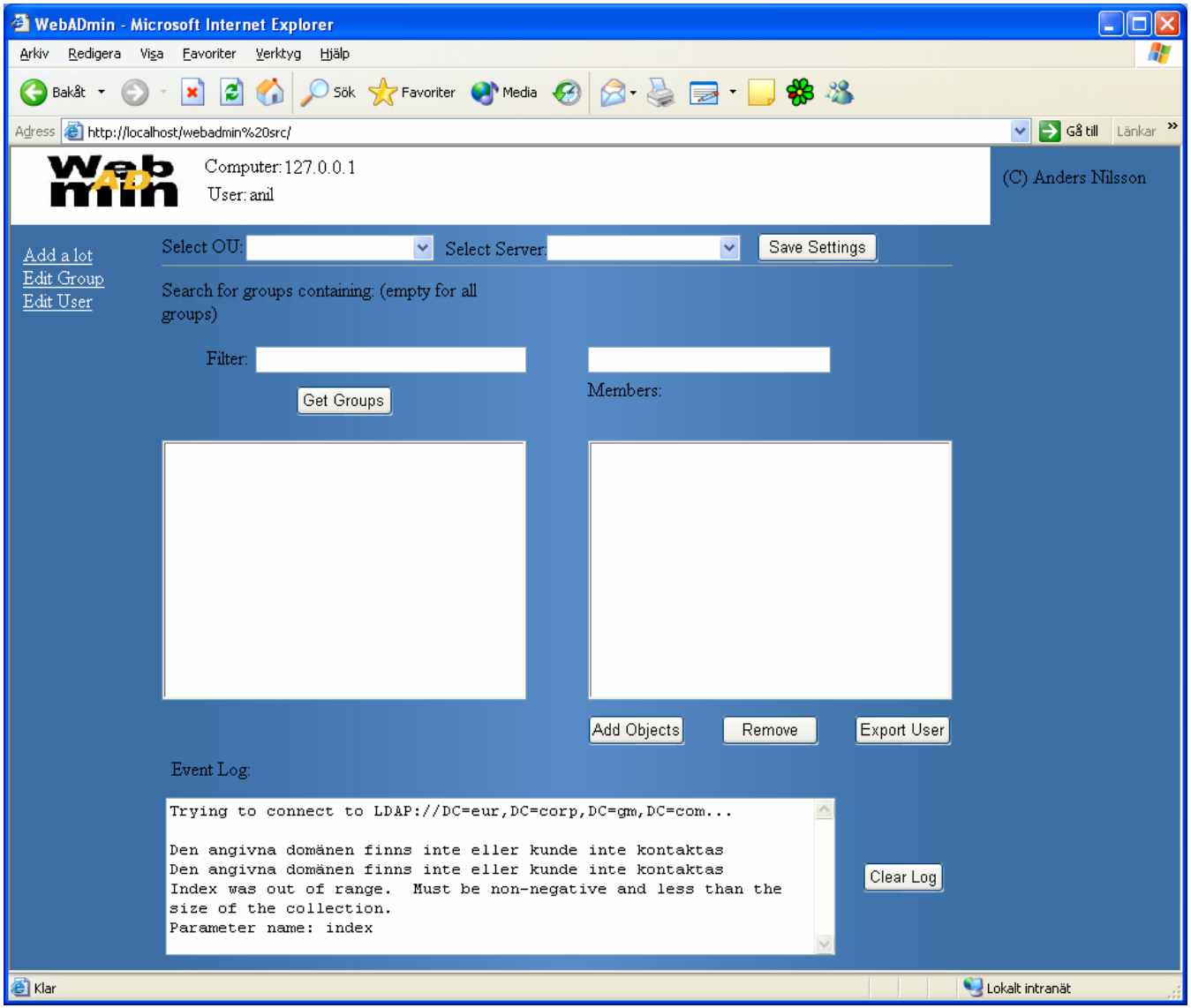
ASP is a technology used to create dynamic web-pages, giving us a tool which we can use to interact with the user, creating custom pages depending on the information the users gives us when accessing the pages. It is possible to make use of COM objects in

ASP pages. This gives us a method with which we can combine a user interface which can be reached through the web and a COM object, which handles the communication with the Active Directory.

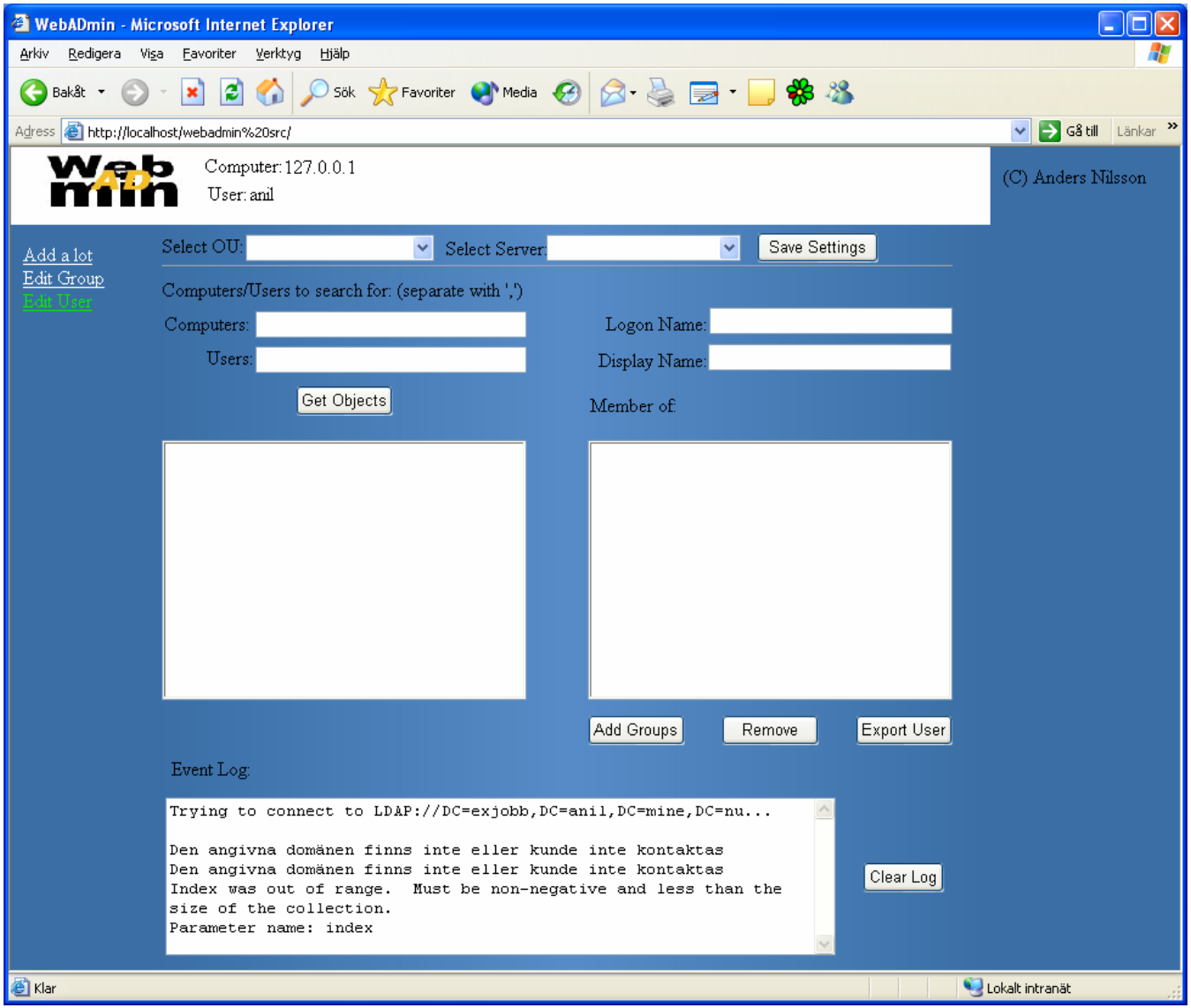
ASP.NET

ASP.NET, or ASP+ as it is also spelled, is not a sequel to ASP 3.0. It has been rebuilt from ground-up to be better and more efficient then ASP 3.0. There are a lot of things in ASP+ that differs from ASP 3.0. Many of them are not relevant to this project, so I leave them and concentrates one the ones that is. One of the most important ability for me is a part named Web Controls. It is part of the .Net framework that enables me to create graphical user interfaces on ASP+-pages that looks just like and functions just like a ordinary Windows application. I short it is .Net's replacement for ActiveX. Another part of ASP+ is DirectorySearcher, which is a class used to communicate with directory services. Actually, it is a wrapper for the ADSI COM-object, available in .Net. To make use of this technology, I have to have an IIS server with the .Net Framework installed to be able to make an ASP.Net page available to a client. The client does not have to have .Net Framework installed though.

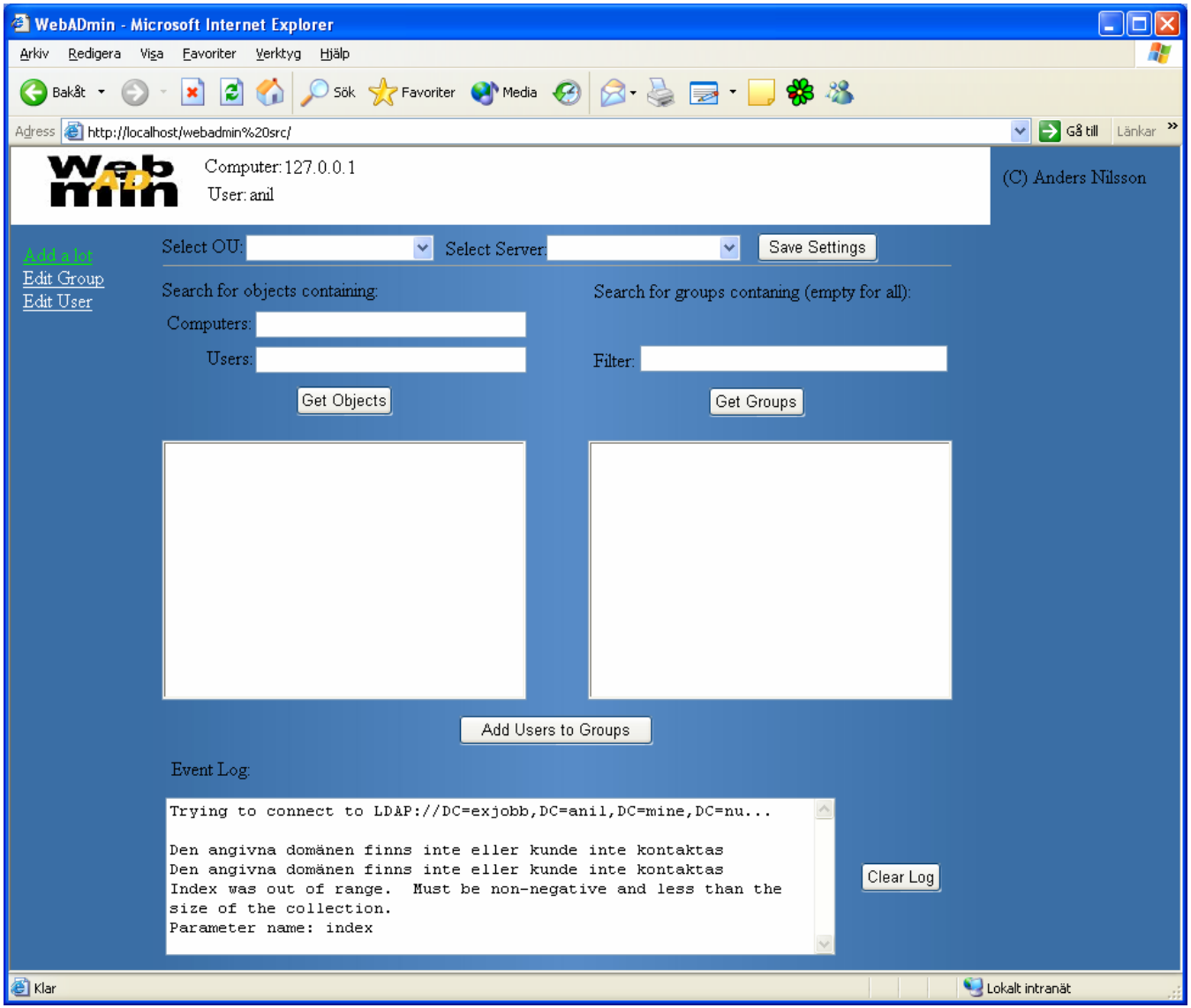
Appendix C
User Interfaces
EditGroup


WebADmin - Microsoft Internet Explorer
Arkiv Redigera Visa Favoriter Verktyg Hjälp
Bakåt Sök Favoriter Media
Adress http://localhost/webadmin%20src/
Gå till Länkar
Computer: 127.0.0.1
User: anil
(C) Anders Nilsson
Add a lot
Edit Group
Edit User
Select OU:
Select Server:
Save Settings
Search for groups containing: (empty for all groups)
Filter:
Get Groups
Members:
Add Objects
Remove
Export User
Event Log:
Trying to connect to LDAP://DC=eur,DC=corp,DC=gm,DC=com...
Den angivna domänen finns inte eller kunde inte kontaktas
Den angivna domänen finns inte eller kunde inte kontaktas
Index was out of range. Must be non-negative and less than the size of the collection.
Parameter name: index
Clear Log
Klar
Lokalt intranät

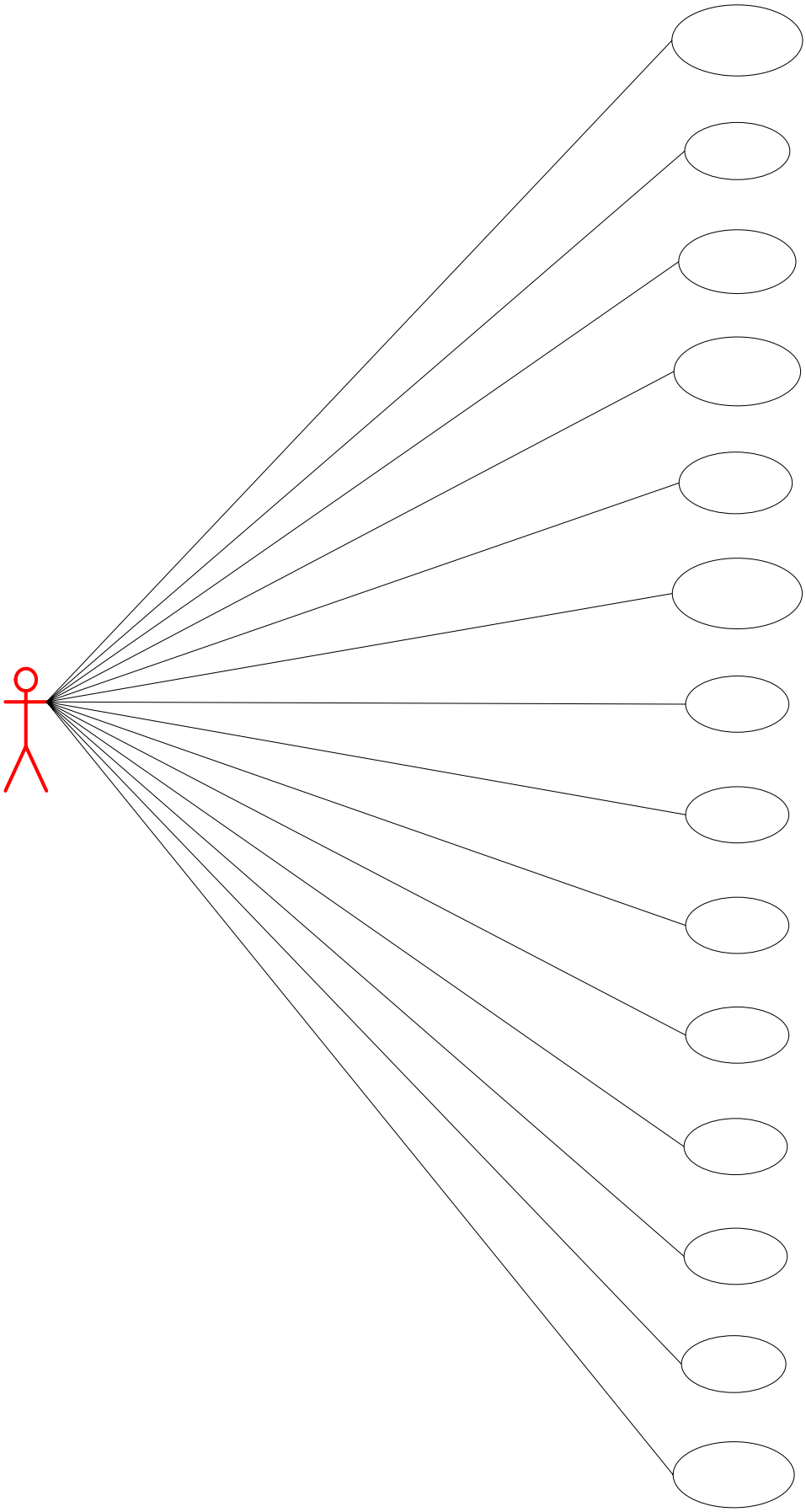
EditUser


WebADmin - Microsoft Internet Explorer
Arkiv Redigera Visa Favoriter Verktyg Hjälp
Bakåt Sök Favoriter Media
Adress http://localhost/webadmin%20src/
Gå till Länkar
Computer: 127.0.0.1
User: anil
(C) Anders Nilsson
Add a lot
Edit Group
Edit User
Select OU:
Select Server:
Save Settings
Computers/Users to search for: (separate with ',')
Computers:
Users:
Logon Name:
Display Name:
Get Objects
Member of:
Add Groups
Remove
Export User
Event Log:
Trying to connect to LDAP://DC=exjobb,DC=anil,DC=mine,DC=nu...
Den angivna domänen finns inte eller kunde inte kontaktas
Den angivna domänen finns inte eller kunde inte kontaktas
Index was out of range. Must be non-negative and less than the size of the collection.
Parameter name: index
Clear Log
Klar
Lokalt intranät

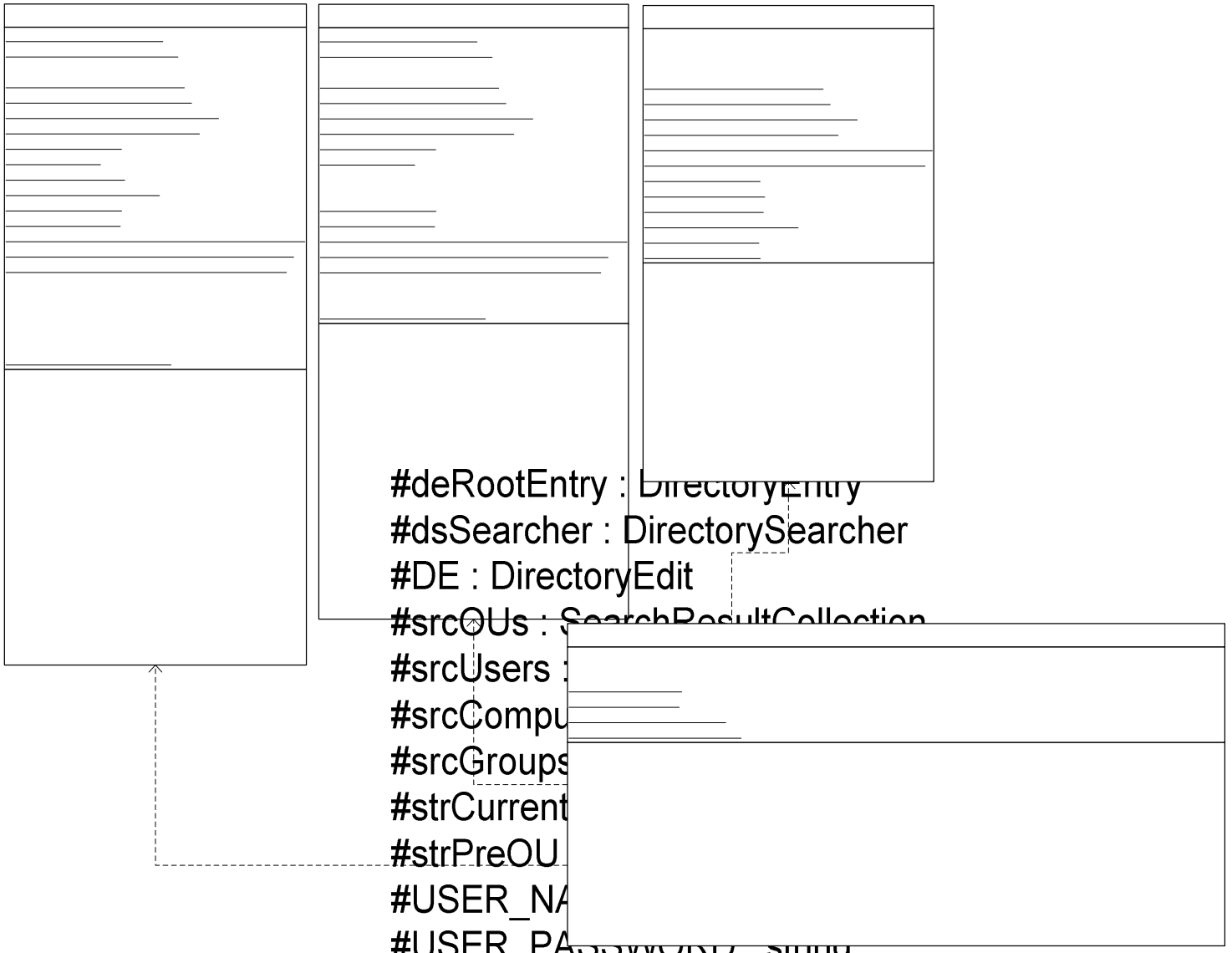
AddMany


WebADmin - Microsoft Internet Explorer
Arkiv Redigera Visa Favoriter Verktyg Hjälp
Bakåt Sök Favoriter Media
Adress http://localhost/webadmin%20src/
Gå till Länkar
Computer: 127.0.0.1
User: anil
(C) Anders Nilsson
Add a lot
Edit Group
Edit User
Select OU:
Select Server:
Save Settings
Search for objects containing:
Computers:
Users:
Get Objects
Search for groups contaning (empty for all):
Filter:
Get Groups
Add Users to Groups
Event Log:
Trying to connect to LDAP://DC=exjobb,DC=anil,DC=mine,DC=nu...
Den angivna domänen finns inte eller kunde inte kontaktas
Den angivna domänen finns inte eller kunde inte kontaktas
Index was out of range. Must be non-negative and less than the size of the collection.
Parameter name: index
Clear Log
Klar
Lokalt intranät

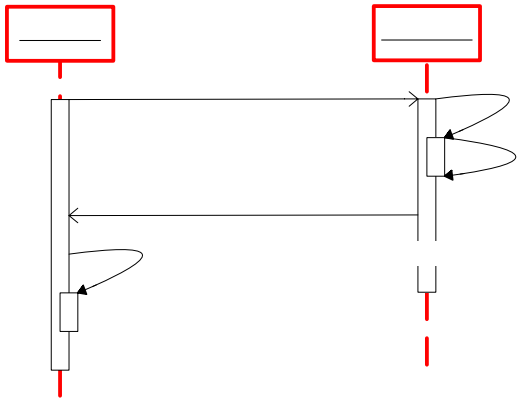
Appendix D
UML Diagrams
UseCase

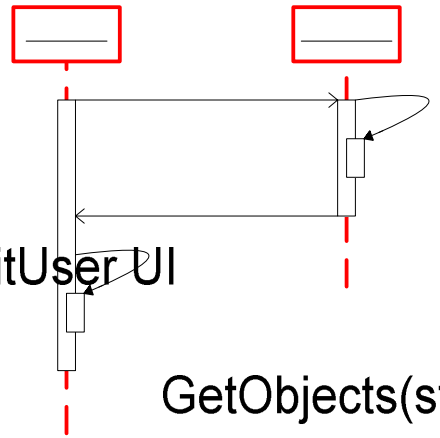


ClassDiagram


#deRootEntry : DirectoryEntry
#dsSearcher : DirectorySearcher
#DE : DirectoryEdit
#srcOUs : SearchResultCollection
#srcUsers :
#srcCompu
#srcGroups
#strCurrent
#strPreOU
#USER_NA
#USER_PASSWORD : string
#SERVER : string = ""
#DOMAIN : string = ""
#scUsersInList : StringCollection = new StringCollection()
#scOUPaths : StringCollection = new StringCollection()
#deRootEntry : Directo
#dsSearcher : Director
#DE : DirectoryEdit
#srcOUs : SearchResu
#srcUsers : SearchRes
#srcComputers : Searc
#srcGroups : SearchR
#strCurrentOU : string
#strPreOU : string
#USER_NAME : string
#USER_PASSWORD
#SERVER : string = ""
#DOMAIN : string = ""
#scGroupsInList : Strin
#scOUPaths : StringCo

Edit User Sequence Diagram Page 1



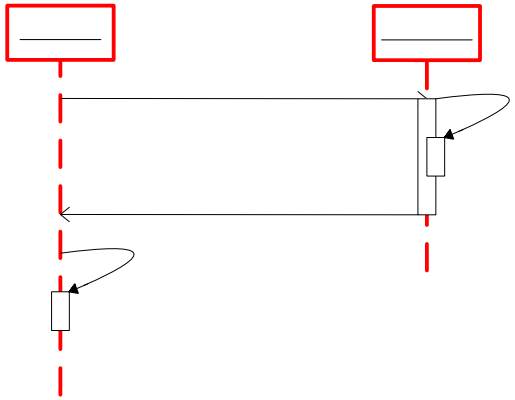


EditUser UI

DirectoryEdit

GetObjects(string):SearchResultCollection

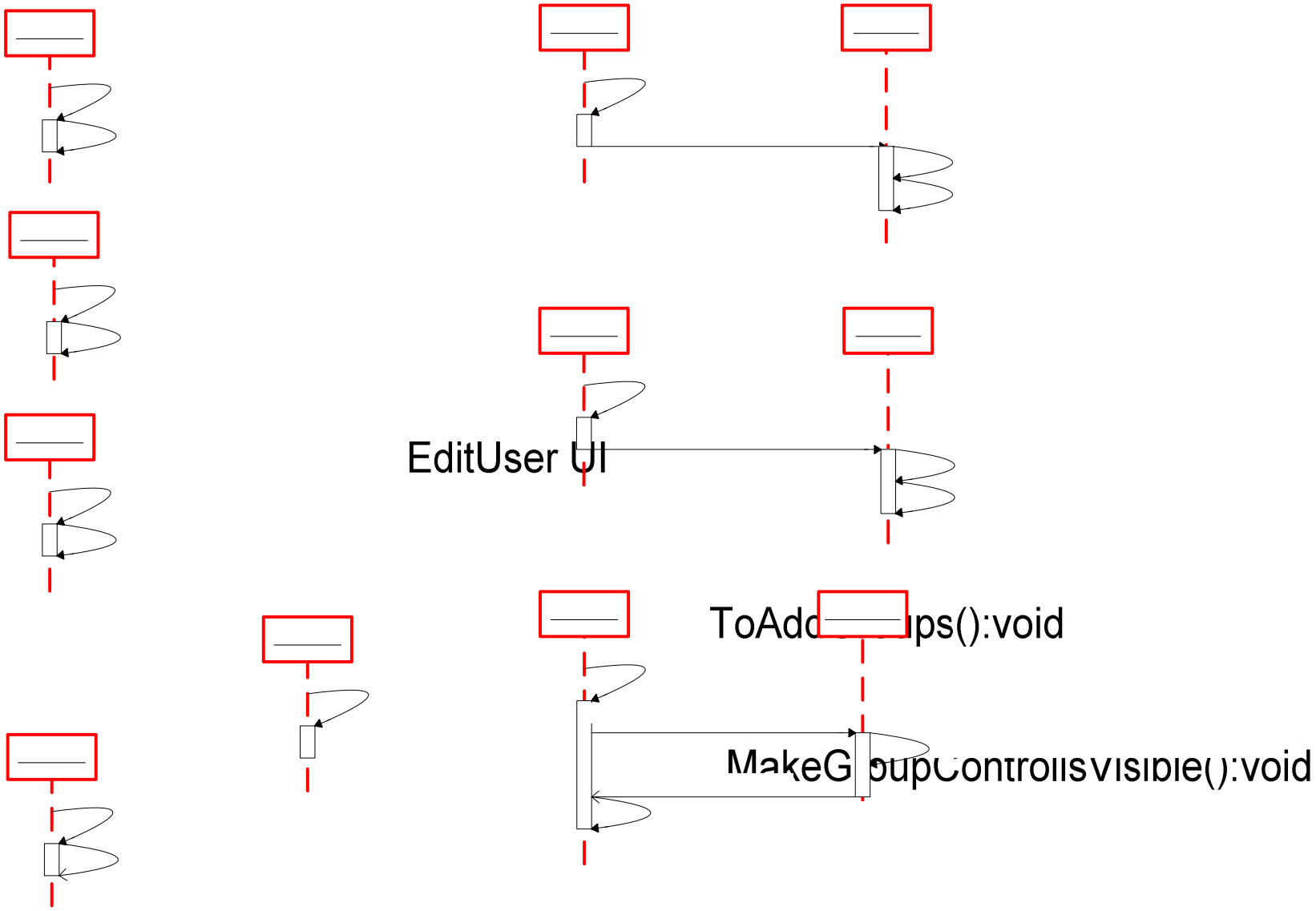
*



SearchResultCollection

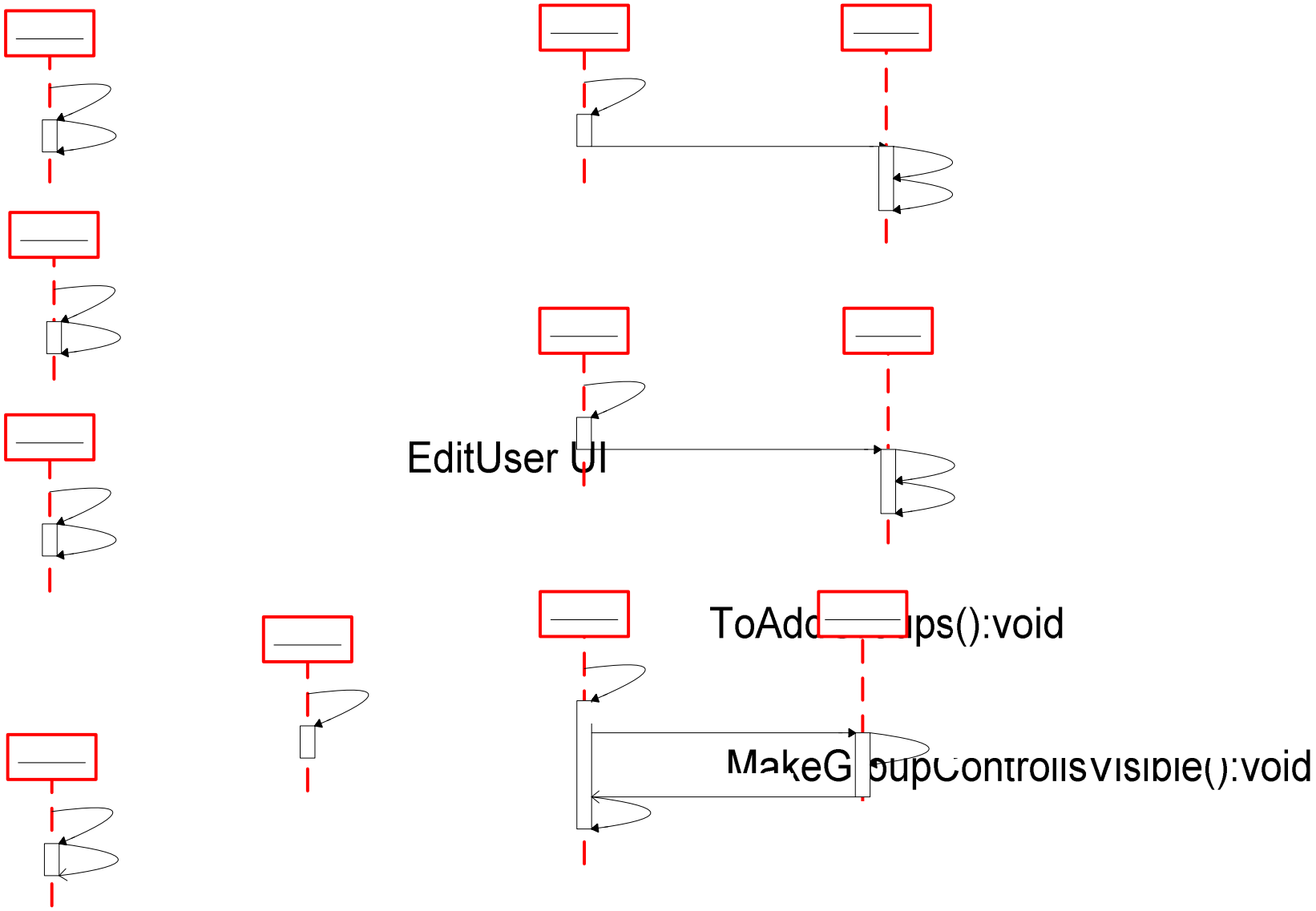
*[for each result in collection]AddNameToListbox():voi

Edit User Sequence Diagram Page 2



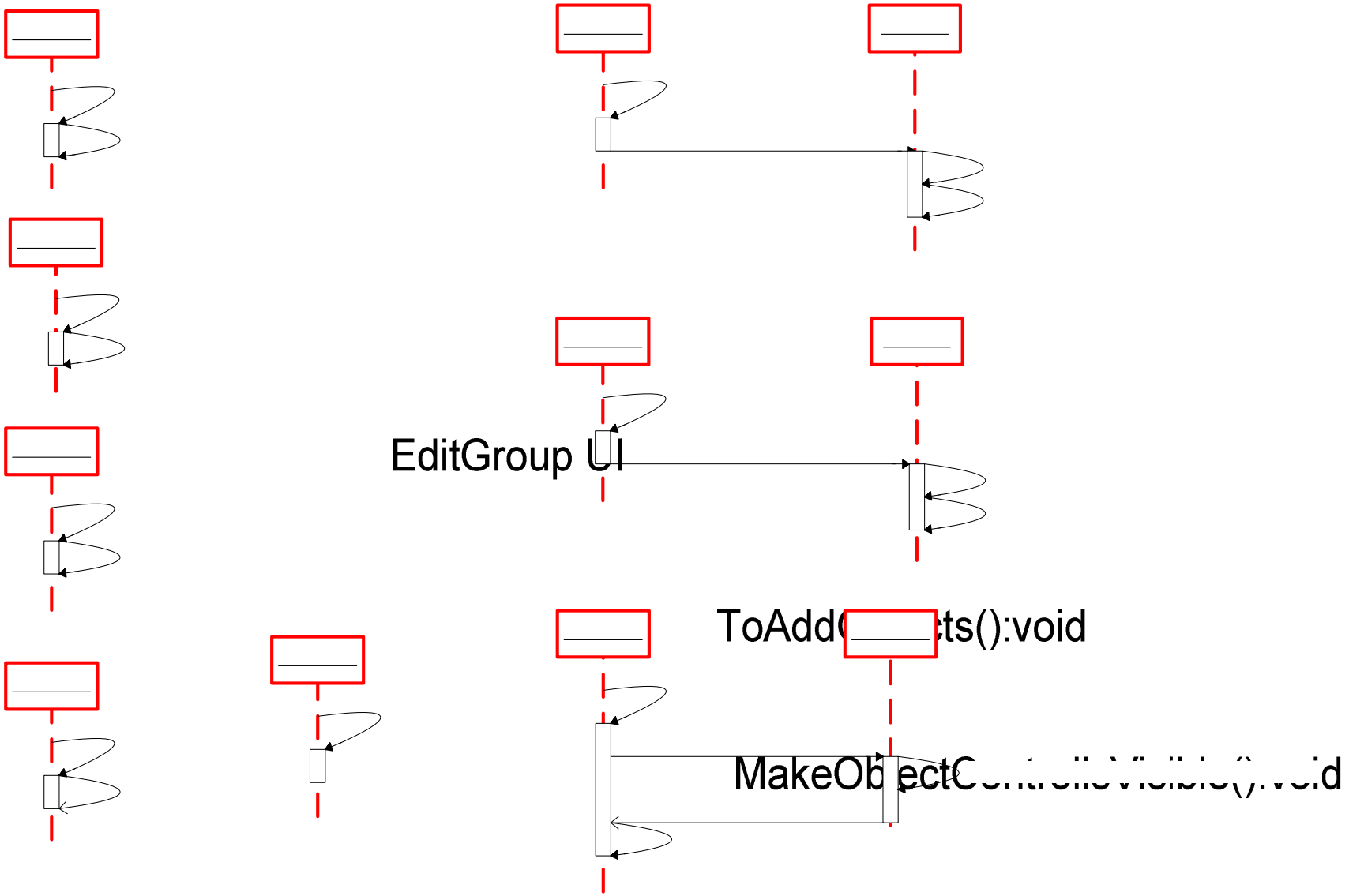
EditUser UI

Edit Group Sequence Diagram Page 1

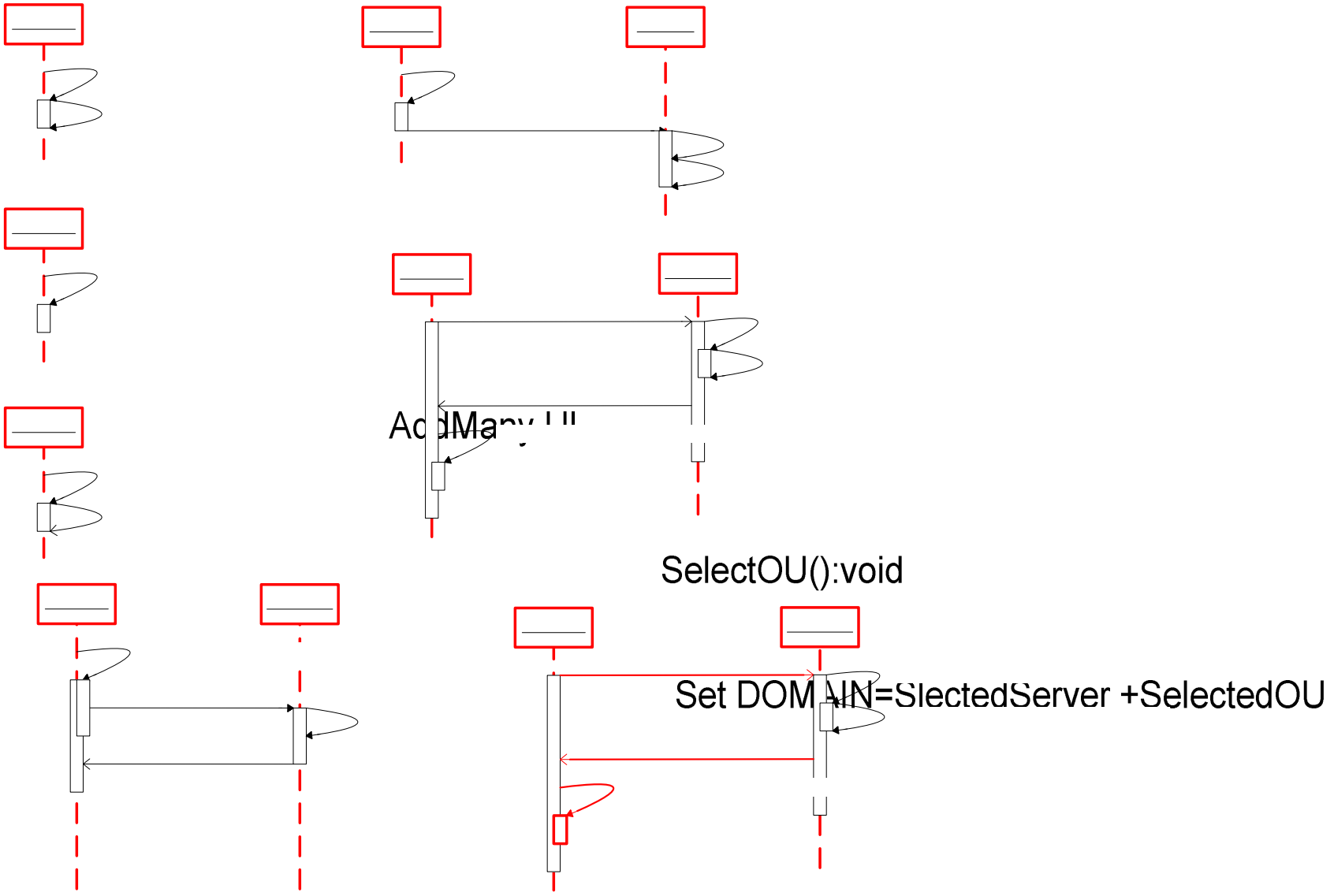


EditUser UI

Edit Group Sequence Diagram Page 2


EditGroup UI
ToAdd
ts():void
MakeOb
ectC
d

Add Many Sequence Diagram


SelectOU():void
Set DOMAIN=SlectedServer +SelectedOU
AddMany UI