

A New Internet Service Provider Billing System

Firas Abdullah Thweny Al-Saedi¹, Shaima' Safaaldin Bahaaldin²

¹Computer Engineering Department, Al-Nahrain University, Baghdad, Iraq

²Computer Center Department, Al-Nahrain University, Baghdad, Iraq

¹firas_alsaidi@yahoo.com

²eng_shaima1183@yahoo.com

Abstract - This paper presents an Internet Service Provider Billing System with unique characteristics and rich feature content, enabling significant cost reduction. The main mechanism of this system is controlling the Online Charging System (OCS). The support for high user load capacity, traffic management, and revenue management makes this system ideal for any small/medium/large ISP. The features of this System are: Tracking, Re-invoicing, Adjustments/credits account creation, verifying and controlling the access time for both LAN and WAN systems in a real time environment. The system includes unlimited scalability, Industry-leading billing capabilities, Web-based administration. It was tested and ran successfully on an Intranet system, that contains a Windows Domain Controller and each PC was a member of that Domain Controller.

Keywords - (OCS) Online Charging System, (LDAP) Lightweight Directory Access Protocol, (ADSI) Active Directory Services Interface, (RADIUS) Remote Authentication Dial In User Service, (ADDC) Active Directory Domain Controller.

I. INTRODUCTION

Over the years, authentication, authorization, and accounting (AAA) has changed dramatically as users of new world access technologies seek a way to authenticate, authorize, and start accounting records for billing user time on their networks. Security for user access to the network and the ability to dynamically define a user's profile to gain access to network resources has a legacy dating back to asynchronous dial access. AAA network security services provide the primary framework through which a network administrator can set up access control on network points of entry or network access servers, which is usually the function of a router or access server. Authentication identifies a user, authorization determines what that user can do, and accounting monitors the network usage time for billing purposes [1].

AAA information is typically stored in an external database or remote server such as RADIUS or TACACS+. The information can also be stored locally on the access server or router. Remote security servers, such as RADIUS and TACACS+, assign users specific privileges by associating attribute-value (AV) pairs, which define the access rights with the appropriate user.

All authorization methods must be defined through AAA. Major information service providers such as Dialog can now be reached via the Internet. While it is relatively simple for an entrepreneur to set up a small computer capable of providing information to the worldwide Internet community, it is much more difficult to arrange a mechanism to charge users for the

services rendered and to collect payments. Current billing mechanisms for electronic services are costly and inconvenient for both service providers and end users. In the absence of a centralized billing service, users must initiate a service agreement with each service before using its services, and must keep track of its access point, password, and bills [2].

Also, there is no central directory of service providers. It is uneconomical for small service providers to advertise, check credit, authenticate users, control access, and keep usage statistics. Both service providers and end users need a reliable, easily accessible, fast and inexpensive intermediary, a billing service [3]

II. BILLING SYSTEM ARCHITECTURE

A. Requirements

There are lots of requirements for a billing system. One of the key requirements is the ease of use. It has to be designed in such a way that even a person with basic knowledge of computers can operate it with relative ease. It should be self-explanatory and able to handle large customer base. The billing system must be generic, database independent, operating system independent, can be expandable in future and designed as a marketable product [4].

An example to one of the ISP Billing Network is shown in Figure 1.

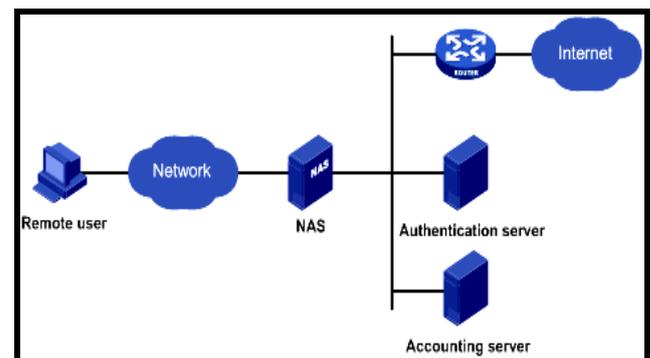


Fig. 1 ISP Billing Network

The billing system must be able to customize different roles for different user needs. It should have a login system. The passwords must be encrypted when transmitted from client to server and vice-versa. The billing system must be accessible using a browser and can provide reports as a collection of important information.

It should be easy to access a customer's information. It means to store and retrieve customer service information. Simultaneous use of this billing system should not be hindrance and all data must be accurately stored; since the nature of a billing system is such that some of the parameters like reports, employee...etc change, and must be editable. The same billing system must be able to run from different locations connected to the same database. Last but not least, the customer information shouldn't be accessible to all other users of this software [5].

B. Technology

The server technology used is the LDAP/ADSI with the Active Server Pages (ASP). Active Directory services use the Lightweight Directory Access Protocol (LDAP) as its core protocol and can work across operating system boundaries, integrating multiple namespaces. It can manage application-specific directories, as well as other NOS-based directories, to provide a general-purpose directory that can reduce the administrative burden and costs associated with maintaining multiple namespaces. Active Directory services are not an X.500 directory. Instead, it uses LDAP as the access protocol and supports the X.500 information model without requiring systems to host the entire X.500 overhead. The result is a high level of interoperability that supports real-world heterogeneous networks.

The information stored within the structure of the Active Directory can be expanded and customized through the use of various tools. One such tool is the Active Directory Services Interface (ADSI), which is available to Windows developers. ADSI provides objects and interfaces that can be accessed from within common programming languages, such as Visual Basic, Visual C++, and Active Server Pages (ASP). This feature allows the Active Directory to adapt to special applications and to store additional information as needed. It also allows all of the various areas within an organization (or even between them) to share data easily based on the structure of the Active Directory [6].

It is important to understand the distinction between logical and physical components of the Active Directory. When planning the objects and domains, then it will be taken into account. Because LDAP is a standard, it also facilitates interoperability between other directory services. Furthermore, communications can be programmed using objects such as the ADSI, which has always been a very effective way of dealing with users in a Windows network. For data transport, LDAP can be used over TCPI/IP, thus making it an excellent choice for communicating over the Internet, as well as over private TCPI/IP-based networks.

The Microsoft ASP is a server-side scripting environment that can be used to create and run dynamic, interactive Web server applications. With ASP, it can be possible to combine HTML (Hypertext Markup Language) pages, script commands, and COM components to create interactive Web pages or powerful Web based applications, which are easy to develop and modify. ASP is easy to program and relatively secure. It provides good database connectivity. It provides

developer productivity, application performance, reliability, and deployment.

One of the challenges faced by Web developers is how to create a coherent application out of a series of independent HTML pages. This problem is a particular concern in Web development because HTTP (Hypertext Transfer Protocol) is a stateless protocol. Each browser request to a Web server is independent, and the server retains no memory of a browser's past requests. The HTTP 1.0 protocol did not provide a mechanism to maintain state information between requests from a browser.

To overcome this limitation, application developers required a technique to provide consistent user sessions on the Web. ASP provides powerful and flexible Web session management that requires no special programming. The Session object, one of the intrinsic objects supported by ASP, provides a developer with a complete Web session management solution. The Session object supports a dynamic associative array that a script can use to store information [7].

III. SYSTEM DESIGN and IMPLEMENTATION

All too often, systems and network administrators implement hardware and software without first taking the time to evaluate the prerequisites. For example, there is no ability to implement a tape backup solution without first ensuring that the appropriate network connectivity and attachment interface are available on servers. Installation and configuration of the Active Directory is no exception. The main physical components that form the basis of the Active Directory are Windows Server 2003 domain controllers. ASP, Visual Basic Script and Visual C#.NET are used as the scripting language for the client/server Model.

The reasons are availability of software, ease of use and familiarity. Figure (2) demonstrates the flowchart of a Billing System Model. The connection to the database is made once for each user session. User accounts are used to enforce the security within the network environment. These accounts define the login information and passwords that are used to receive permissions to network objects. Computer objects allow systems administrators to configure the functions that can be performed on client machines throughout the environment.

Both User accounts and Computer objects enable security to be maintained at a granular level. Although security can be enforced by placing permissions directly on User and Computer objects, it is much more convenient to combine users into groups. For example, if there are three users who will require similar permissions within the Accounting department, it can be placing all of them in one group.

If some users are removed or added to the department, it can be easily make changes to the group without having to make any further changes to the security permissions. The proper use of groups assists greatly in implementing and managing security and permission within the AD.

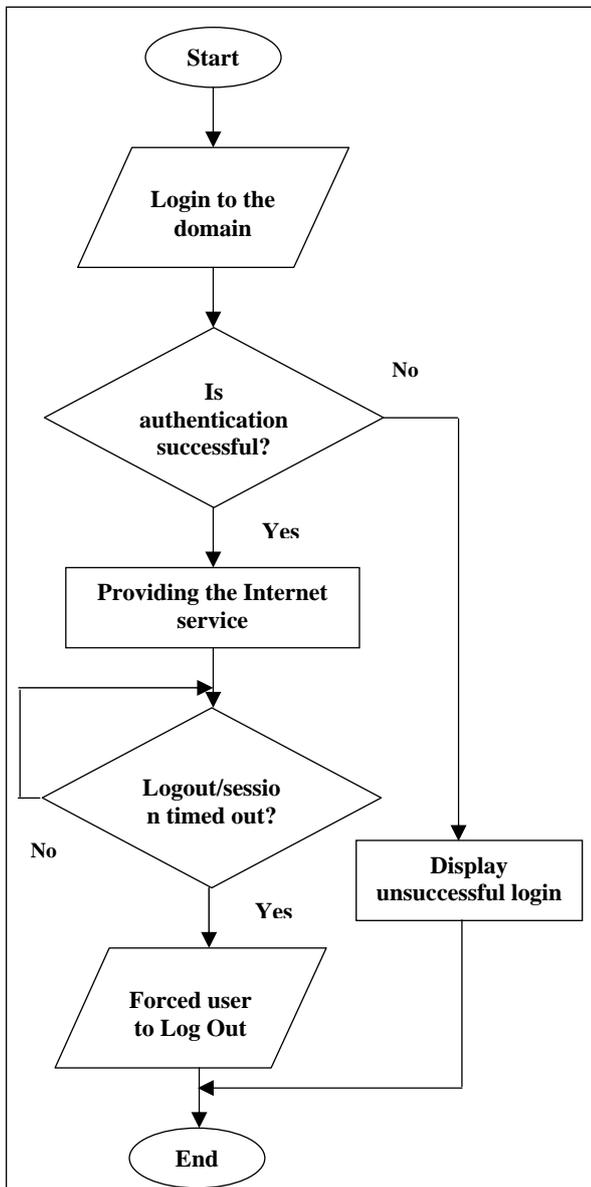


Fig. 2 Flowchart of a billing system model

A. Registration Process

In this section a registration web site was designed to register the user in the Active Directory Database using the Authentication Code and Serial Number written at the Prepaid Internet Card. Figure (3) represents the source code of this process and its flowchart in Figure (4).

First of all, the user should have login to the domain to be able to start registering by using the default user for Domain Login (User Name: nunv@MyCompany.com and Password: Test123try); this default user has a limited Internet time enough for registration or refilling the Account. After login to the domain, the Registration Web Site will be available for the user as the Home Page to start registering at any browser, by filling the Information; which are the Authentication Code, Serial Number (The available SecondaryOU Accounts in the Domain Controller), and the proposed User name with the password (The New PrimaryOU Accounts).

```

private void Button1_Click(object sender, System.EventArgs e)
{
    string strAuth= txtCode.Text;
    string strSerial= txtNum.Text;
    string strLogin="MyCompany.com"+"@"+"\ "+strAuth;
    DirectoryEntry entry=new
    DirectoryEntry(null,strLogin,strSerial); try{ //Bind to //the
    Native AdsObject to force Authentication
    Object obj=entry.NativeObject;
    DirectorySearcher search=new DirectorySearcher(entry);
    search.Filter="(samAccountName="+strAuth+")";
    search.PropertiesToLoad.Add("cn");
    SearchResult result=search.FindOne();
    if(null!=result)
    {
        string strLog="MyCompany.com"+"@"+"\ "+strAuth;
        DirectoryEntry domain= new
        DirectoryEntry(null,strLog,strSerial);
        Object objc=domain.NativeObject;
        DirectorySearcher search2= new DirectorySearcher(domain);
        Search2.Filter="(samAccounName="+strAuth+")";
        Search2.PropertiesToLoad.Add("cn");
        SearchResult result2=search.FindOne();
        If(null!=result2){DirectoryEntry obUser= new
        DirectoryEntry(result2.Path);
        DirectoryEntry newPath= new
        DirectoryEntry("LDAP://OU=PrimaryOU,
        DC=MyCompany, DC=com");
        obUser.MoveTo(newPath);string strUser=txtName.Text;
        string strPSW=txtPassword.Text;
        obUser.Rename("cn="+strUser);
        obUser.Properties["displayName"].Value=strUser;
        obUser.Properties["userPrincipalName"].Value=strUser;
        obUser.Properties["samAccountName"].Value=
        strUser;obUser.Properties["description"].Value=36000;
        obUser.CommitChanges();Object
        obRet=obUser.Invoke("SetPassword",strPSW);
        Status.Text="Congratulations! You are accepted";}}
    }
}
  
```

Fig. 3 The registration process source code

After Authentication, the Billing System will delete the existed account in the SecondaryOU (i.e. Authentication Code and Serial Number); in order that no one will use this card again as a second-hand card, and create new accounts in the PrimaryOU (i.e. User Name and Password) and finally assign a time script to the logged user with the specified time given by the Prepaid Internet card to limit the domain access time.

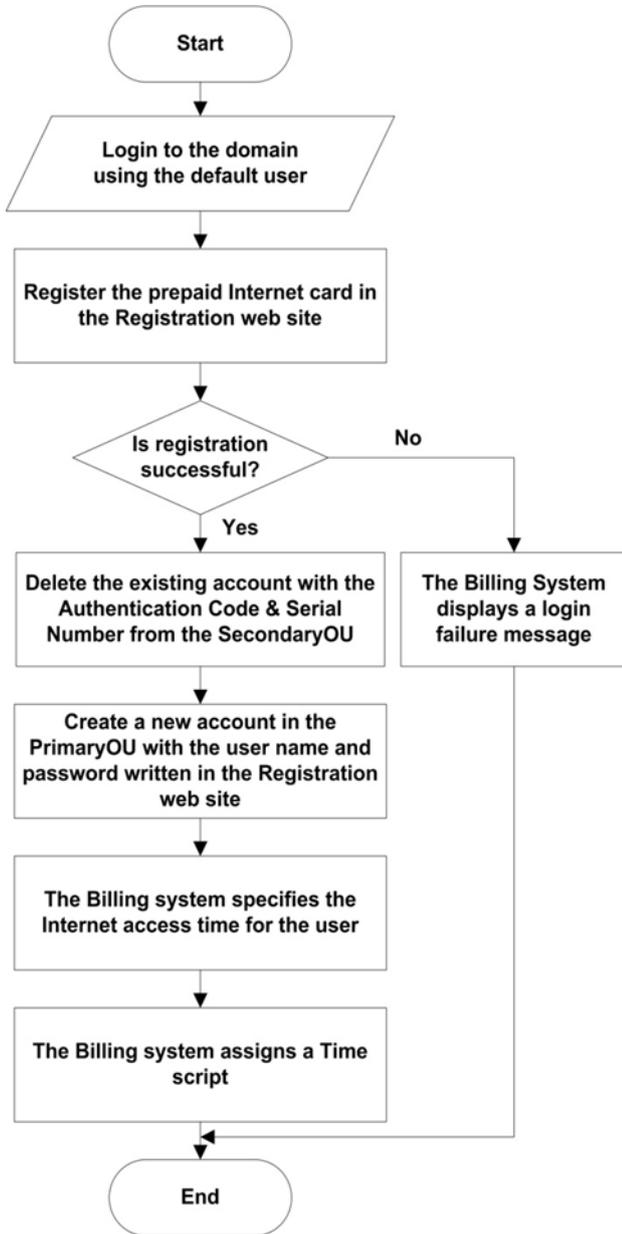


Fig. 4 The flowchart of the registration process

Script policies are specific options that are part of Group Policy settings for users and computers. These settings direct the operating system to the specific files that should be processed during the logon/logoff process. It can be possible to create the scripts themselves by using the Windows Script Host (WSH) or by using standard batch file commands. WSH

is a utility included with the Windows Server 2003 operating system Domain Controller.

The time script will be initialized whenever a login occurred in the system, in order to track the user access time whenever using the Internet service, thus when the user Internet access time has expired, then a force logoff dialog box appear as shown in Figure (5) and the source code of the time script shown in Figure (6).



Fig. 5 The logout process dialog box

```

Dim ads, objUser, iUserTime, sUserTime, iDescr,
Dim oShell
Set ads=createObject ("ADSystemInfo")
StrUser=ads.Username
Set objUser= GetObject("LDAP:///" & strUser)
Do
Wscript.Sleep 1000
sDescr=objUser.Get("description")
'Convert the string description to integer description
iDescr=CInt(sDescr)
iUserTime=iDescr-1 sUserTime=cStr(iUserTime)
objUser.put "description", sUserTime
objUser.SetInfo if (iUserTime=0) Then
objUser.AccountDisabled=True
objUser.SetInfo
Dim WshShell
Set
WshShell=WScript.CreateObject("WScript.Shell")
oShell.Popup "Save your work immediately!", 60,
"LoggingOff",vbInformation+vbSystemModal
WshShell.run "shutdown /f "
WScript.Quit End if Loop
  
```

Fig. 6 The time script source code

B. Refill Account Process

In this section, a web page is designed for refilling the account of the user using Visual C#.NET as shown in Figure (7). The user will type the Authentication Code and Serial Number; which are written in the prepaid Internet Card, with the user name and password, then pressing the Refill account button for refilling the account. If a new user did not register initially, and open the Refill account site to fill the information, then a login failure message will appear, because this account does not already exist at the domain controller and the user needs to register first, not to refill an account that does not exist yet at all.

This process is similar to the process when the user goes to the Internet service provider asking for a reconnection and pays the money to get the Internet service. Figure (8) demonstrates the refilling account site process as a flowchart.

C. Check Account Process

In order to check the existing account, the user should open the Check Account web site and follow the instructions on the displayed page, so if the user exists, then the Billing System will reveal the remaining time for the user to use the Internet service. The Check account site process source code is shown in Figure (9) and the flowchart shown in Figure (10) demonstrate the whole process.

IV. RESULTS and DISCUSSIONS

Using the ASP technology and Visual C#.NET language, three web pages was created for the ISP Billing System. If a new user needs a registration to use the Internet service, then the following steps should be done:

1. Buy a new Prepaid Internet card for the desired hours which contains the Authentication Code and Serial Number.
2. Connect to the Internet either by a wired or wireless connection, and login to the domain using the default account as written at the prepaid Internet Card, such as typing nunv@MyCompany.com as shown in Figure (11).
3. The registration web site will appear as a home page for any browser to sign up for a new account using the registration site, so if it is successful then a congratulation message will appear as shown in Figure (12). After the successful registration, the user will be able to use the Internet service.
4. If there is a need to check for the remaining time for the user, the following site will be used: <http://MyCompany.com/AccountCheck/WebForm1.aspx> then following the instructions displayed on the page, so if the user exists, the user's remaining time will appear as shown in Figure (13).
5. The refill account site will be used for refilling the allowable user time for using the Internet service and typing the Authentication Code and serial number, which are written on a new Internet card, with the user name and password. After completing the refilling account process,

the total user remaining time will appear as shown in Figure (14).

6. Whenever an account has been expired, it will be disabled within the Domain Controller as shown in Figure (15), then the expired account message will be displayed at the check account web site, as shown in Figure (16)

```
private void Button1_Click(object sender, System.EventArgs e)
{
    string strUser= txtName.Text; string strPSW= txtPSW.Text;
    string strLogin="MyCompany.com"+@"\"+strUser;
    DirectoryEntry entry=new
    DirectoryEntry(null,strLogin,strPSW);
    try{//Bind to the Native AdsObject to force Authentication
    Object obj=entry.NativeObject;
    DirectorySearcher search=new DirectorySearcher(entry);
    search.Filter="(samAccountName="+strUser+")";
    search.PropertiesToLoad.Add("cn");
    SearchResult result=search.FindOne();
    if(null!=result){ string strAuth=txtCode.Text;
    string strSerial=txtNum.Text;
    string strLog="MyCompany.com"+@"\"+strAuth;
    DirectoryEntry domain=new DirectoryEntry(null,strLog,strSerial);
    Object objc=domain.NativeObject;
    DirectorySearcher search2=new DirectorySearcher(domain);
    search2.Filter="(samAccountName="+strAuth+")";
    search2.PropertiesToLoad.Add("cn");
    SearchResult result2=search.FindOne();
    If(null!=result2){DirectoryEntry objDelete=new
    DirectoryEntry(result.Path);
    DirectorySearcher objUser=new DirectoryEntry(result2.Path);
    String
    sDescr=objUser.Properties["description"].Value.ToString(); int
    iDescr=System.Convert.ToInt16(sDescr);
    int TotalTime=iDescr+36000;
    string strTime=System.Convert.ToString(TotalTime);
    objUser.Properties["description"].Value=strTime;
    objUser.CommitChanges(); int hours=TotalTime/3600; int
    remainder=TotalTime%3600; int minutes=remainder/60;
    int seconds=remainder%60; Status.Text="Congratulations! You
    Total Time="+hours.ToString()+ "hours,
    "+minutes.ToString()+ " minutes and
    "+seconds.ToString()+ "seconds";}else{
    Status.Text="Sorry! You are not authorized.";}}}
```

Fig. 7 The refill account process source code

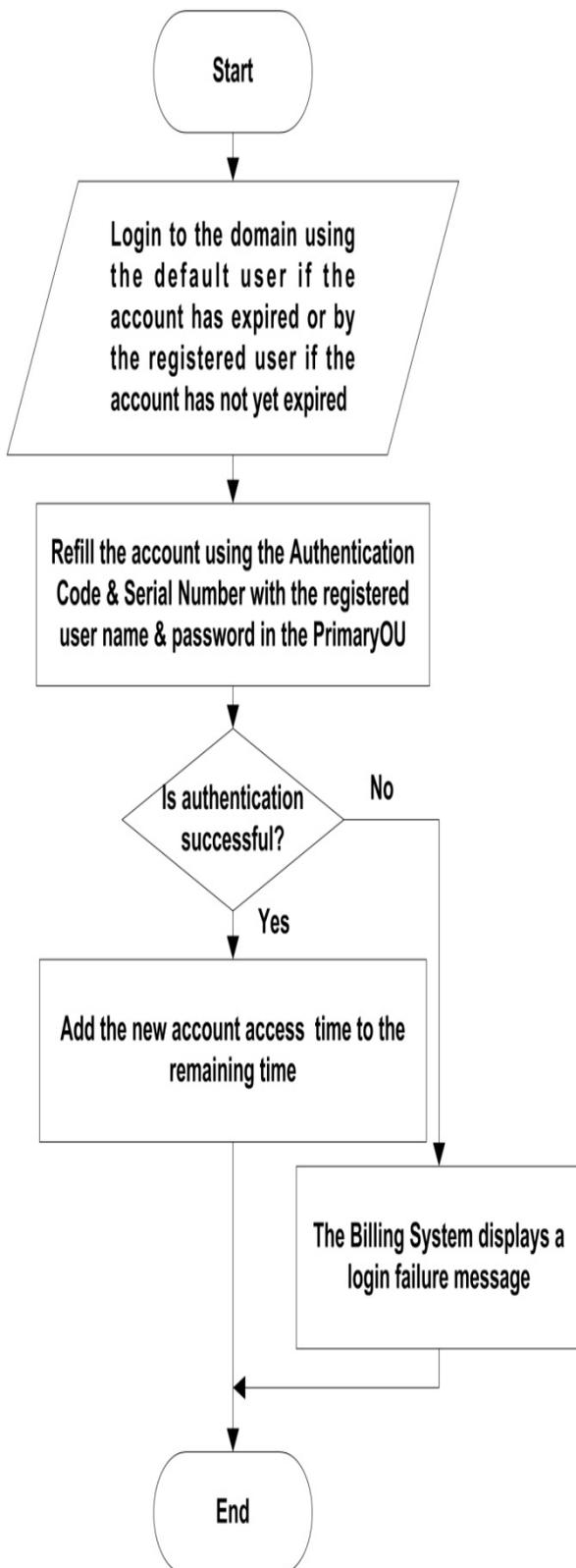


Fig. 8 The flowchart of the refill account process

```

private void Button1_Click(object sender,
System.EventArgs e)
{
string strUser= txtName.Text;
string strPassword= txtPSW.Text;
string strLogin="MyCompany.com"+@"\"+strUser;
DirectoryEntry entry=new
DirectoryEntry(null,strLogin,strPSW);
Try
{
//Bind to the Native AdsObject to force
Authentication
Object obj=entry.NativeObject;
DirectorySearcher search=new
DirectorySearcher(entry);
search.Filter="(samAccountName="+strUser+")";
search.PropertiesToLoad.Add("cn");
SearchResult result=search.FindOne();
if(null!=result)
{
DirectoryEntry objUser=new
DirectoryEntry(result.Path);
string descr=(string)
objUser.Properties["description"].Value;
int time= System.Convert.ToInt16(descr);
int hours=time/3600;
int Remainder=time%3600;
int minutes=Remainder/60;
int seconds=Remainder%60;
Status.Text="Time
remaining="+hours.ToString()+" hours, "+
minutes.ToString()+" minutes and
"+seconds.ToString(); }}}
  
```

Fig. 9 The Check account process source code

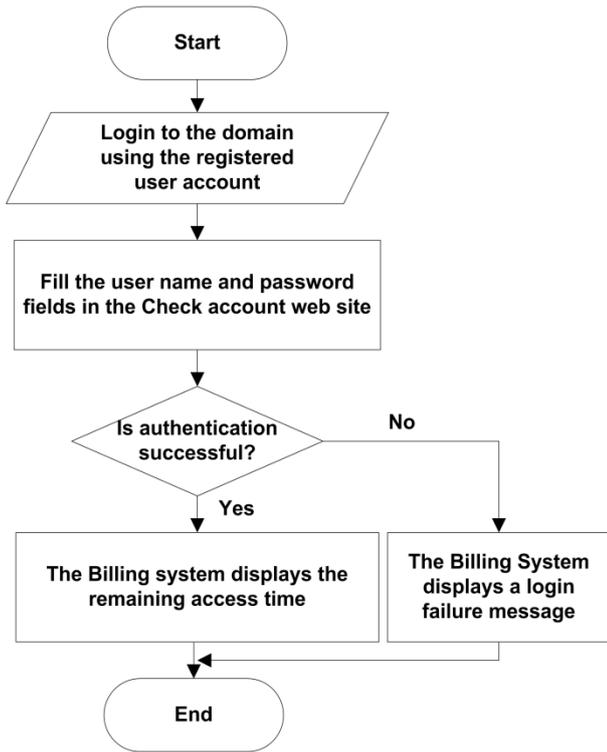


Fig. 10 The flowchart of the check account process

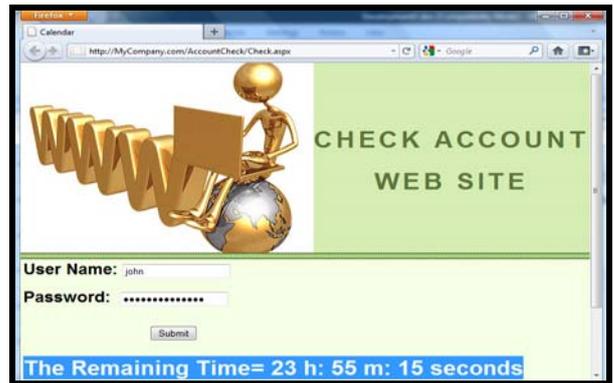


Fig. 13 The account check web site



Fig. 14 The refill account web site

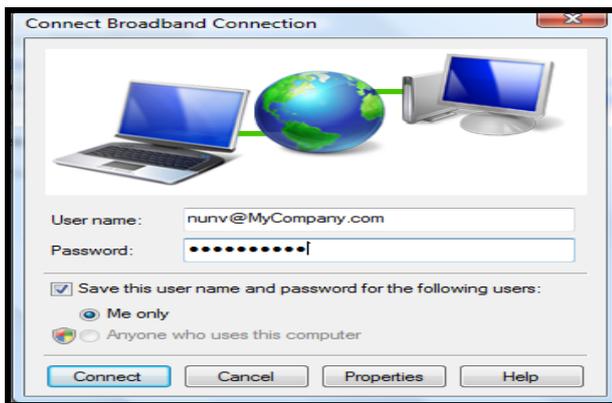


Fig. 11 The domain logging window

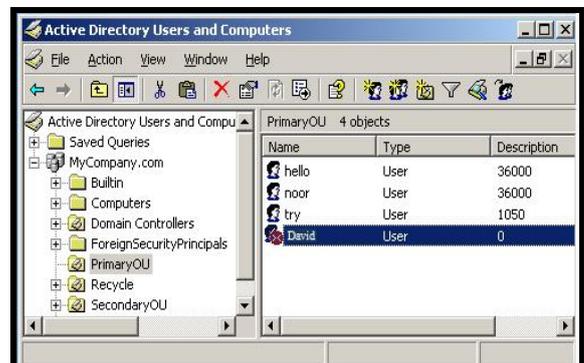


Fig. 15 The disabled account of the user

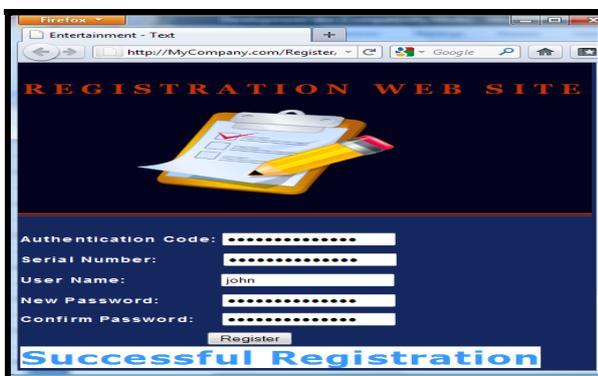


Figure 12 The registration for a new account



Fig. 16 The expired account of the disabled user

V. CONCLUSIONS

This paper addresses a Billing problem that occurred in RadControl, Alepo and Nomadix Billing System companies. For example in a RadControl system, when a user signs up for a full hour Internet access, the Billing system starts its deduction once login to the system. In other words, the Billing system starts its mission and finishes even if the user does not use the Internet service at all. Therefore, this system was designed to overcome this issue by the secure web sites and time script in a real time environment. The Time script tracks the user access time, so that when the user registers for 24 hours (i.e. $24*60*60=86400$ seconds) of the Internet Service, and starts using it for only 5 minutes, then the Internet service will still be available for 23 hours and 55 minutes (i.e. 86100 seconds) and for a whole year, certainly depending on the user usage. In addition, the ISP Billing System designed with secure sites; hence the number of characters in the password field are 14, in order to increase the number of possibilities for predicting the password by the hackers, also the number of account failed are 2; which represents the number of the failed trials to enter the password, with increasing the Account Lockout duration to 30 minutes.

However, there is no possibility that a hacker try to use a second-hand card, because once the user registers the card in the billing system, some information will disappear for

verification, and the registration process will certainly be failed, because of the missing information which were used automatically at the registration of the actual user and removed after that. It means that the registration page checks some information before registering the user, thus this process reduces the hacking process. Finally, the time script forces the user to logoff in order to end the user session when the Internet Service time has expired.

REFERENCES

- [1] Marshall B. Romney and Paul J. Steinbart, *Accounting Information Systems*, USA: Prentice Hall, 2011.
- [2] Ulric J. Gelinias, Richard B. Dull and Patrick Wheeler *Accounting Information Systems*, South-Western College Pub, 2011.
- [3] Cynthia Richards, *Coding Basics: Medical Billing and Reimbursement Fundamentals*, Delmar Cengage Learning, 2009.
- [4] Donald E. Kieso, Jerry J. Weygandt and Terry D. Warfield, *Intermediate Accounting*, Wiley, 2009.
- [5] Emphasis Charles T. Horngren, George Foster, Srikant M. Datar, Madhav Rajan, Chris Ittner, *Cost Accounting: A Managerial*, Prentice Hall, 2009.
- [6] Anne Boehm and Joel Murach, *Murach's ASP.NET 4 Web Programming with C# 2010 (Murach: Training & Reference)*, Mike Murach & Associates, 2011.
- [7] Imar Spaanjaars, *Beginning ASP.NET 4: in C# and VB Wrox Programmer to Programmer*, Wrox, 2010.