CLINICAL TRIALS *DATA MANAGEMENT AND TRIAL CONDUCT*

Clinical Trials 2009; **6**: 378–385

Turning *Access*™ into a web-enabled secure information system for clinical trials

Dongquan Chen[a,b,c], *Wei-Bang Chen*[a], *Mayhue Soong*[b], *Seng-Jaw Soong*[a] and *Helmuth F. Orthner*[c,†]

Background Organizations that have limited resources need to conduct clinical studies in a cost-effective, but secure way. Clinical data residing in various individual databases need to be easily accessed and secured. Although widely available, digital certification, encryption, and secure web server, have not been implemented as widely, partly due to a lack of understanding of needs and concerns over issues such as cost and difficulty in implementation.

Purpose The objective of this study was to test the possibility of centralizing various databases and to demonstrate ways of offering an alternative to a large-scale comprehensive and costly commercial product, especially for simple phase I and II trials, with reasonable convenience and security.

Methods We report a working procedure to transform and develop a standalone *Access*™ database into a secure Web-based secure information system.

Results For data collection and reporting purposes, we centralized several individual databases; developed, and tested a web-based secure server using self-issued digital certificates.

Limitations The system lacks audit trails. The cost of development and maintenance may hinder its wide application.

Conclusions The clinical trial databases scattered in various departments of an institution could be centralized into a web-enabled secure information system. The limitations such as the lack of a calendar and audit trail can be partially addressed with additional programming. The centralized Web system may provide an alternative to a comprehensive clinical trial management system. *Clinical Trials* 2009; **6**: 378–385. http://ctj.sagepub.com

Nomenclature

AD	Active Directory. AD is the directory service used in a Windows-based server and provides the foundation for Windows-based distributed networks.
ASP	Active Server Pages, which enable a web page to run scripts and access databases.
CA	Certificate Authority. The main function of a CA is to issue, revoke, and mange certificates that ensure the identification of both a server and a client.
CTL	certificate trust list.
CRL	certificate revocation.
DNS	Domain Name System, a system that translates Internet domain names (such

[a]Biostatistics and Bioinformatics Unit, Division of Preventive Medicine and Center for Clinical & Translational Science, Univ. of Alabama at Birmingham (UAB). Birmingham, AL, USA, [b]Clinical Studies Unit, UAB Comprehensive Cancer Center. Birmingham, AL, USA, [c]Department of Health Services Administration, School of Health Professions, UAB. Birmingham, AL, USA

Author for correspondence: D. Chen, Medical Towers 636, 1717 11th Avenue South, Birmingham, Alabama 35205. Tel: (205) 934-3376, Fax: (205) 934-4262. E-mail: dongquan@uab.edu

[†]Dr Helmuth F. Orthner (1941–2009) passed away after completion of the manuscript. The authors wish to express their deep appreciation for his contribution to the project and condolences to his family.

10.1177/1740774509338228

as uab.edu into IP numbers such as 138.26.1.1.

FTP File transfer protocol

HIPAA Health Care Insurance Portability and Accountability Act.

https Secure Socket Layer (SSL) over Hypertext Transfer Protocol (https). https protocol enables the secured transmission of Web pages.

IIS Internet Information Server. IIS is Microsoft's Web server that runs on Windows NT platforms.

IP internet protocol that defines the way data is delivered over the networks.

ODBC Open Database Connectivity. ODBC is a standardized interface, or middleware, for accessing a database from a program.

RAS Remote Access Server or Service. A RAS system enables users to log into a local area network using a modem, a wireless card, etc.

SSL Security Socket Layer, a commonly used protocol for managing the security of a message transmission over the Internet.

Access™ *and* Windows™ are trademarks of Microsoft.

Background

Fully functional databases can be expensive to either purchase or develop, which makes their use difficult or impossible for some small studies. Our group tested open-source systems such as OpenClinica (openclinica.org/), TrialDB (ycmi.med.yale.edu/trialdb), and more recently, Cancer Central Clinical Participant Registry (C3PR, // cabig.nci.nih.gov/tools/c3pr) of Cancer Biomedical Informatics Grid™ (CaBIG) program of the National Cancer Institute (NCI). These open source systems are less expensive than commercial systems, but a lack of customer support is a major drawback. Commercial systems such as Oracle Clinical SiteMinder™, TrialMinder™, and Velos (www.velos.com) cost more and need special programming. Implementation of these systems is time consuming and costly. They may not be suitable for small organization with limited resources and/or with limited number of trials especially phase I, II, and multicenter trials.

This creates a dilemma for investigators lacking the necessary resources or who cannot justify the expense associated with the use of such databases. Under these circumstances, a common approach is to develop a series, often fragmented, of databases to accommodate the assortment of data needed for the study. Consequences may include inadequate security, difficulty in data access and data sharing across databases. Other issues derived from these databases include a lack of standard terminology and difficulties in querying, reporting, and database management [1].

This indicates, therefore, a need for a simple, secure, cost-effective, and easy-to-set-up system. We tested this possibility by enabling Web accessibility to a Microsoft Access™ database and applying additional security measures such as digital certifications [2–4]. The web-based trial system is important especially for multi-center trials since it allows data collection from various departments or institutions over the web [5]. We report here a procedure to transform a standalone database into a web-enabled system with reasonable functionalities including the enhanced security by applying digital certification, a way to authenticate both users and machines.

Methods

We installed a standalone authentication system using Microsoft Certificate Authority (CA) that is a part of Windows™ 2003 server for user identification; an Active Directory (AD) for user account management; and a Web server using Microsoft Internet Information Server (IIS) for the Web interface. The system architecture is shown in Figure 1. The certificate authority issues digital certificates to client computers, server computers, and the CA itself. An additional Remote Access Server (RAS) is implemented to allow remote access [6]. A connection between clients and the database was created through a method for client-database connection (Open Database Connectivity, ODBC). The secured connection was created by enabling Secure Socket Layer (SSL) over Hypertext Transfer Protocol (https). The detailed installation procedure is in the Appendix.

Results

System design and management

The databases are located on two Intranet servers and accessible by various departments and programs. We centralized databases by consolidating data from four individual *Access*™ databases on campus. As shown in Figure 1, we developed the integrated system for data collection, user authentication, server authentication, and secure network communication. To access the database, a client must have a digital identification (certificate 3), a user account, and a secure connection. The authentication server is trusted among clients and servers.

System interfaces, implementation, and work flow

We designed the data entry gateways for various entities within the University (Figure 2), for protocol-related information (Figure 3), and for commonly used reports (Figure 4). We tried to minimize the changes needed for the interfaces within the forms. Database query and reporting over the web follow the same scheme as using the standalone *Access*™ database. The sample code for the programming is in the Appendix.

Cost evaluation for a similar system

To setup and maintain the system, one person with a background in Computer Sciences is required

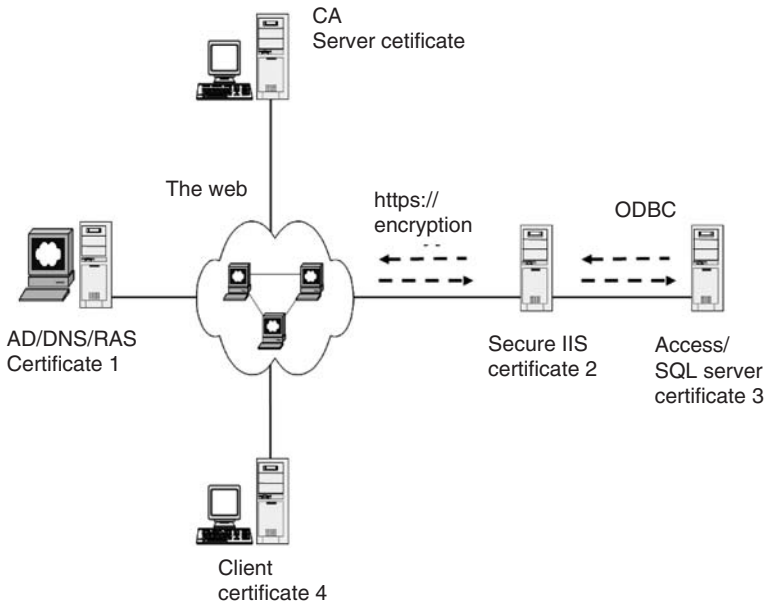

Figure 1 Network architecture for a secure information system. The AD with a DNS controller enables clients/users to access the database through RAS and web interface

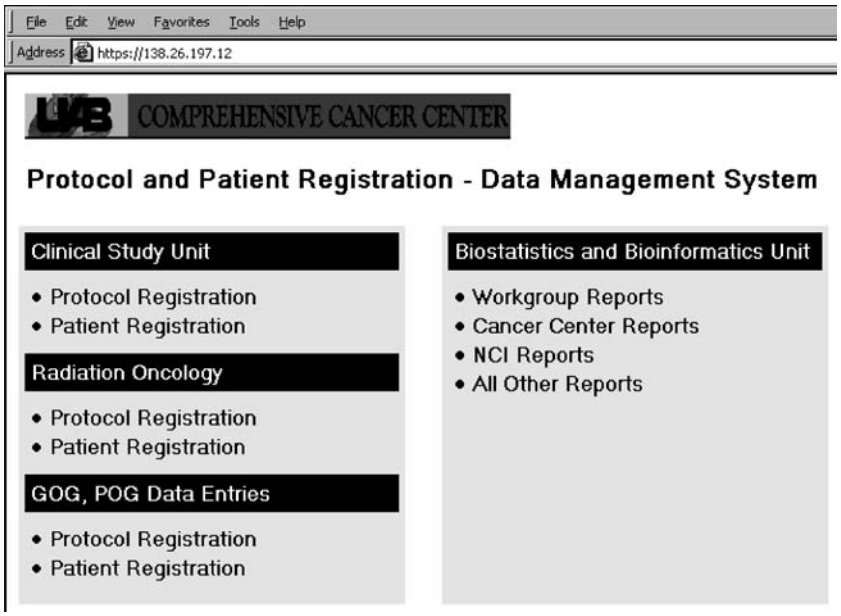

Figure 2 A web-enabled clinical trial database for data entry and reporting. The web-enabled system uses *Access*-derived forms over the web. Authorized users access data entry forms and reports over the web

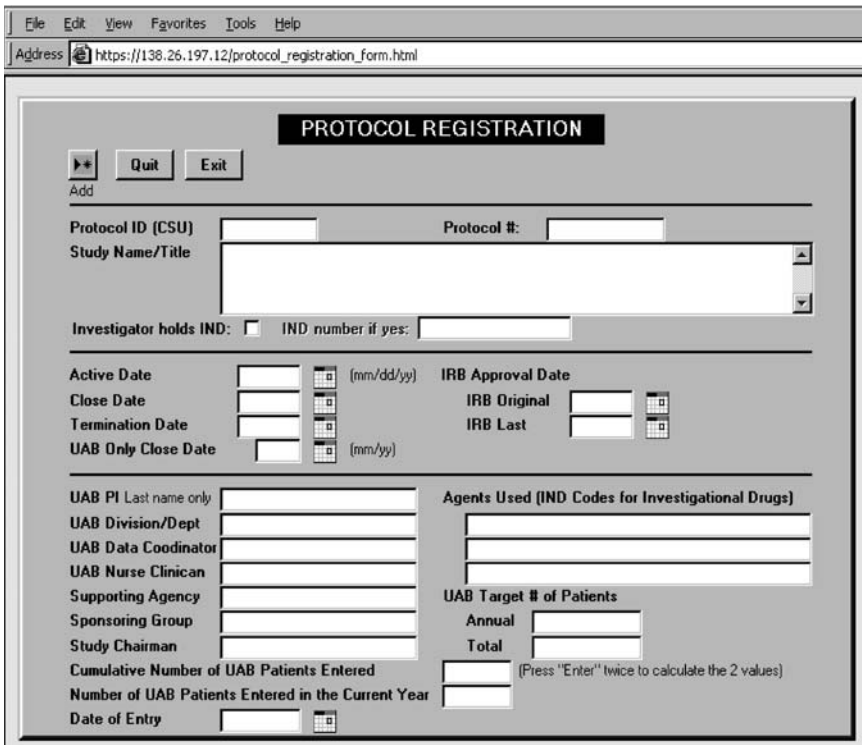

Figure 3 The web-based data entry for protocols. The protocol registration form is one of many forms used to collect protocol-related information. The collected data was inserted into a table of the database through SQL query and active server pages

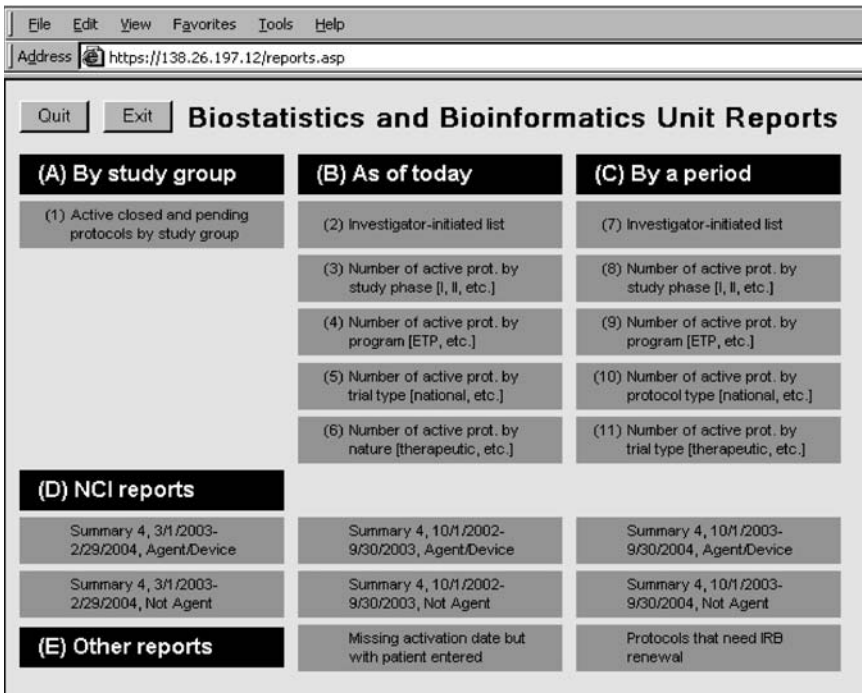

Figure 4 The query-based report gateway. The report section leads to both local and national reports that may require certain information in a fixed format. Mort reports are in read-only Portable Document Format (PDF) format

Table 1 Cost to implement a Web-enabled secure *Access*™ database*

Components	No. needed	Unit price	Total US$
Server hardware	3	2000	6000
Server/database software	4/2	500	3000
Software licensee fees	4	500	2000
Personnel /training	1	45,000/year	45,000
			56,000

*Based on the cost in 2008.

with part-time effort. The system can be an alternative to those organizations that cannot afford costly commercial products such as Oracle Clinical SiteMinder™, TrialMinder™ (oracle.com), and Velos™ (velos.com) systems. The direct cost for hardware, software, and personnel is around US $56,000 based on 2008 prices as shown in Table 1.

Discussion

There is a need for a simple and easy-to-set-up system for data entry, queries, and reporting. Our data is collected from various departments or programs on the campus. Centralization of these databases offers not only Web-based data collection but also Web-based data entry, query, and reporting.

Digital certification to secure web-based data entries and reporting

The secure connection and secure server are widely used for sensitive network transactions. The secure connection is achieved through encryption and SSL technology. The authentication of both server and clients is achieved by application of the digital certificates. The advancement of encryption and web-based technologies makes it possible to enter, retrieve, and transmit data over the Internet with reasonable security [7], for a multi-center clinical trial [8], for example. The scripting language may enable these functions to collect trial data and provide a sophisticated user interface [9,10]. In our study, we have successfully applied the technology for the same purpose.

Implementation, maintenance, and administration

The system we developed may not need regular programming for maintenance although the constant change of data elements requires changes in data entry forms and in report designs – especially for multicenter clinical trials [11–14]. Other organizational issues such as ownership of both data and the centralized system need to be addressed during implementation.

System limitations

The lack of functions such as calendar and audit trails is the major disadvantage of using an *Access*™ database for clinical trials. Although not the focus of the study, these issues need to be addressed by additional software add-ins, modules, or programming. The creation of a temporary table, for example, for all changes after initial data entry, may be a partial solution to the lack of an audit trial.

Conclusions

The clinical trial databases scattered in various departments of an institution could be centralized into a web-enabled information system. We report an easy-to-follow procedure to transform a standalone database such as an *Access*™ database into a secure web-based information system. Although it has certain limitations such as the lack of a calendar and audit trail, it could function, with additional programming, as an alternative to a comprehensive clinical trial management system.

Acknowledgments

The authors would like to thank colleagues in the Biostatistics and Bioinformatics Unit for helpful discussions, Mrs Laura Gallitz for proofreading, and Drs O. Dale Williams and Alan Cantor for helpful discussions. This project has been funded in part with US federal funds from the National Cancer Institute, NIH under CA-13 148 and from the NCRR, NIH under 1UL1RR025777.

References

1. **Rossille D, Burgun A, Pangault-Lorho C** *et al*. Integrating clinical, gene expression, protein expression and preanalytical data for in silico cancer research. *Stud Health Technol Inform* 2008; **136**: 455–60.
2. **Georgiadis CK, Mavridis IK, Pangalos GI.** Healthcare teams over the Internet: towards a certificate-based approach. *Stud Health Technol Inform* 2002; **90**: 184–88.
3. **Georgiadis CK, Mavridis IK, Pangalos GI.** Healthcare teams over the Internet: programming a certificate-based approach. *Int J Med Inform* 2003; **70**(2–3): 161–71.
4. **Zuckerman AE.** Restructuring the electronic medical record to incorporate full digital signature capability. *Proc AMIA Symp* 2001; 791–95.

5. **Arlet V, Shilt J, Bersusky E** *et al*. Experience with an online prospective database on adolescent idiopathic scoliosis: development and implementation. *Eur Spine J* 2008; **17**(11): 1497–506.
6. **Chen D, Soong SJ, Grimes GJ** *et al*. Wireless local area network in a prehospital environment. *BMC Med Inform Decis Mak* 2004; **4**(1): 12–20.
7. **Sierdzinski J, Karpinski G.** Electronic patient record and archive of records in Cardio.net system for telecardiology. *Pol J Pathol* 2003; **54**(3): 223–26.
8. **Kuchenbecker J, Dick HB, Schmitz K** *et al*. Use of internet technologies for data acquisition in large clinical trials. *Telemed J E Health* 2001; **7**(1): 73–76.
9. **Kelly MA, Oldham J.** The Internet and randomised controlled trials. *Int J Med Inf* 1997; **47**(1–2): 91–99.
10. **Nadkarni PM, Marenco L.** Easing the transition between attribute-value databases and conventional databases for scientific data. *Proc AMIA Symp* 2001; 483–87.
11. **Carr MA.** Coordinating data management for multiple ongoing clinical trials and registries. *Top Health Rec Manage* 1990; **11**(2): 13–19.
12. **Duftschmid G, Gall W, Eigenbauer E** *et al*. Management of data from clinical trials using the ArchiMed system. *Med Inform Internet Med* 2002; **27**(2): 85–98.
13. **Fulcher SF, Burris TE.** A computerized recall system for clinical trials. *Ann Ophthalmol* 1988; **20**(1): 10–16.
14. **Gassman JJ, Owen WW, Kuntz TE** *et al*. Data quality assurance, monitoring, and reporting. *Control Clin Trials* 1995; **16**(2 Suppl): 104S–136S.

Appendix

This is a technical manual for installing the **AD**, **IIS**, **CA**, **SQL** server and typical coding for **ASP**.

Windows Active Directory (AD)-based user-account management

Setup procedure and settings:

(1) Install Windows™ 2000 or 2003 (**W2K**) Server in the **AD** computer as shown in **Figure 1**.
(2) Install an **AD** with a **DC** by typing and running code dcpromo after following Start/Run window. The DC will function as a **DNS** server. Repeating the procedure will uninstall it, if desired. A valid domain such as clinical trials.ad.uab.edu is needed and must be registered at parent domain such as uab.ad.edu. The **AD** manages all user accounts for accessing the Internet Information Server (**IIS**) and backend database.

Microsoft IIS as the Web server

Setup procedure and settings:

(1) Install **W2K** Server in the secure IIS computer as shown in the **Figure 1.** The machine will be used as a web server and provide web interface for data entry.
(2) Install IIS by following Add/Remove Programs/Add/Remove Windows Components/Internet Information services (checked).
(3) Allow desired user account for remote access.
(4) Turn off the file transfer protocol (**FTP**) server and simple mail transfer protocol (SMTP) service by unchecking the options before installing or stopping the services after installations.
(5) Disable the web-based administration console for the **IIS**.
(6) Request and install the server certificate from the self-managed **CA**, as described below.
(7) Enable secure communications by requiring clients **SSL** and 128 bit encryption for accessing the secure server.
(8) Setup and enable certificate trust list (**CTL**) and add the server certificate to the list.
(9) Create a certificate revocation list (**CRL**) to further enhance the security.
(10) Issue all authorized users the client certificates from the SMCA in order to access the database through the Web server.
(11) Map Clients' certificates to their **AD/DC** user accounts to simplify the user login.

Microsoft Access™ and/or SQL server

Setup procedure and settings:

(1) Install Microsoft Access 2000 or later version as a database application. It will be used for data entry, storage and reporting.
(2) Design forms for data collection. The forms include protocol registration, patient registration. For both reporting and billing purposes, the data integrity should be reinforced while changes in one table will be propagated into another table. This also simplifies the database maintenance procedure. The relational data model should be applied where primary and secondary keys link all tables for being queried as a single data source. Considering the size of the database of any medical center, we think Microsoft Access™ XP or 2003 would be sufficient for data storage and management. More effort should be directed to other aspects of the database such as accessibility, availability, network security, data encryption and transmission, etc.
(3) Install **SQL** server in the same (**IIS** computer) or a separate machine with certificate 3 as shown in the **Figure 1.** It will be used as a backup server for the Access database.
(4) Connect **SQL** server with **IIS** by setting up an **ODBC** connections. Open **ODBC** Data Source Administrator by following the directory in

Control Panel/Administrative Tools / Data Sources / **ODBC**. Add the new **ODBC** connection as a system Data Source Name (**DSN**) and point to the desired database. Add password protection for login ID and password if desired.

Microsoft Certificate Authority (CA)
Setup procedure and settings:

(1) Install Certificate Services by following Add/ Remove Programs/Add/ Remove Windows Components/Certificate Service (checked).
(2) Install **W2K** Server in the **CA** computer as shown in the **Figure 1.**
(3) Set the **CA** standalone or under an enterprise domain (e.g., clinicaltrials.ad.uab.edu).
(4) Set issuing certificate manually (not the automatic issuing by default). This allows the administrator to check the certificate application and user's domain account before issuing the certificate. The certificate can later be mapped to the user account to enhance the added security and simplify the logon procedure.
(5) Set up, as an administrator, a **CTL** for those trusted and a **CRL** for those revoked certificates.
(6) Install the sever certificate (encrypted by using SHA-I algorithm) as a root certificate, and thus trusted among all certificates issued by this **CA**. This can also be achieved by installing a **CA** certification path after login onto **CA** Web-interface.
(7) Enable the server certificate for all purposes including client authentication, secure email, time stamps, and smart card logon if wireless access is desired.

Microsoft ASP and Web-based data entry

We used ASP to access the data source and manually code a password. The Web-based applets will run within the HTML Web-browser to connect to the database for data entry and query purposes and is not visible to clients.

Example code within the **ASP** for data entry:

```
%>
Dim Connec
Set Connec=Server.CreateObject(''ADODB.
Connection'')
CN.Open''DSN=xxx;
uid=yyy;password=zzz;''
Set rsInsertInfor=Server.Create
Object("ADODB.Recordset")
sqlQuery="INSERT INTO infor_patient
(trial_infor) values (''+trial_data+'')"
Connec.open sqlQuery, rsInsertInfor
Set rsInsertInfor=Nothing
Set Conne=Nothing
%>
```

where Dim Connec is to define a variable called Connec that is an object name created in the server. DSN is data source is named xxx, uid user ID, and user password zzz. 'rsInsertInfor' is a record set name that is set to insert data by running a query named 'sqlQuery'. The result is to insert into a table named 'infor_patient' one piece of information (trial_infor) with value=trial_data. The query statement can be elongated to include as many pieces of information with corresponding values into different columns of the same record /a row in a table.

For data retrieval purposes, the procedure is the same except for the sql statement where 'SELECT trial_data FROM infro_patient WHERE...' will be used instead of the INSERT statement. To filter the data, a condition can be set after 'WHERE'. The retrieved data can be formulated to what is desired by end users.

The user account such as sa for administrators or any other account can be used to open connection. User account name and password plus digital certificate must be supplied in order to connect to the database through the IIS.

Query-based reporting over the Web

Query-based reporting can be achieved through the following codes, which pass the values collected through the web interface to the variables within Access queries.

```
Option Compare Database
Dim macroName As String
Dim P1 As String
Dim P2 As String
Public Sub WebAccessHandler(macroName, P1,
P2)
    Dim query_Num As Integer
    Dim query_Name As String
    Dim QD As QueryDef
    Dim old_Query As String
    Dim new_Query As String
    Dim target_1 As String
    Dim target_2 As String
If P1="" Or P1=Null Or P2="" Or P2=Null
Then
    DoCmd.RunMacro (macroName)
Else
    Select Case macroName
    Case "DQ_invesitgatorInitiatedTime
    Window"
      query_Name="DQ_
      invesitgatorInitiated"
    Case "DQ_noOFprotBYnatprotTime
    WindoSum"
      query_Name="DQ_noOFprotBYnatpr-
      otTimeWindo"
```

```
Case "DQ_noOFprotBYphaseTimeWindoSum"
    query_Name = "DQ_noOFprotBYphase
    TimeWindo"
  Case"DQ_noOFprotBYprogramTime
  WindoSum"
    query_Name = "DQ_noOFprotBYprogra-
    mTimeWindo"
  Case "DQ_noOFprotBYtypeTime
  WindoSum"
    query_Name = "DQ_noOFprotBYtype
    TimeWindo"
End Select
P1 = "#" + P1 + "#"
P2 = "#" + P2 + "#"
target_1 = "[start date (mm/dd/yy):]"
target_2 = "[end date (mm/dd/yy):]"
For Each QD In Application.DBEngine(0).
Databases(0).QueryDefs
  If query_Name = QD.Name Then
    old_Query = QD.SQL
    new_Query = QD.SQL
    new_Query = Replace(new_Query,
    target_1, P1)
    new_Query = Replace(new_Query,
    target_2, P2)
    QD.SQL = new_Query
    DoCmd.RunMacro (macroName)
    QD.SQL = old_Query
    Exit For
  End If
 Next QD
End If
End Sub
```

The arguments of the subroutine map to all variables from web. The subroutine retrieves the macroName from the first argument and determines whether there exists any parameter other than macroName passed from the web. If no other variables are passed, the subroutine will run the macro according to the macroName variable; if other variables are received, the subroutine will modify the query by memorizing the original query and replacing the target variables with the values from user, run the specify macro and recover the original query after executing the macro.