DYNAMIC WEB PUBLISHING

– ANALYTICAL REPORT

BY

DUAN SHENG

Bachelor of Science

Jiangsu International Trade Institute

Nanjing, China

1985

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
August, 2001

DYNAMIC WEB PUBLISING

– ANANYTICAL REPORT

Thesis Approved:

Thesis Advisor

Hailin Zhang

J Chandler

Alfred Carlozzi

Dean of the Graduate College

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere appreciation to my major advisor, Dr. G. E. Hedrick, for his intelligent supervision, constructive guidance, encouragement and patience. My sincere gratitude also goes to Dr. H. L. Zhang for giving me the opportunity to complete this "real world" project. I wish to give thanks to Dr. J. P. Chandler and Dr. K. M. George, for their encouragement, help and friendship.

My special thanks extends to my husband, Dr. Letong Tang, my son Dacheng Tang and my parents Dr. Tongsheng Sheng and Dongming Zou, for their support, encouragement, understanding and love.

Finally, I would like thank all the faculty and staffs in Department of Computer Science of Oklahoma State University for their advises and help.

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION 1

1.1 Background 1

1.2 Problem Statement 1

1.3 The Objectives 5

II. LITERATURE REVIEW 7

2.1 Client–Server Computing 7

Two-tier client - server architecture 7

Three-tier client - server architecture 8

Middleware 9

2.2 Data Access 10

Object Linking and Embedding Database (OLE DB) 10

ActiveX Data Object (ADO) 10

2.3 Markup Language 12

Standard General Markup Language (SGML) 12

HyperText Markup Language (HTML) 12

eXtensible Markup Language (XML) 13

2.4 Dynamic Web Page 14

Common Gateway Interface (CGI) 15

Active Server Page (ASP) 16

Chapter Page

III. APPLICATION REQUIRMENTS AND

DEVELOPMENT TOOLS..18

3.1 Business Requirement Analysis..18

3.2 Selection of Platform and Implementation technology.......................19

IV. IMPLEMENTATION..24

4.1 General 24

4.2 Longin..28

4.3 Select 29

4.4 Report..38

4.5 Others Important Functions..48

V. SUMMARY AND FUTURE WORK...61

5.1 Summary...61

5.2 Future Work..62

APPENDIX

Acronyms...63

Chemical Elements, Compounds..65

Bibliography...66

LIST OF FIGURES

Figure Page

1. SWFAL current information system..2

2. SWFAL new information system 5

3. A two-tier client-server model..8

4. A three-tier client-server model..9

5. Web publishing with ASP technology..22

6. A three-tier client-server architecture
implemented with ASP..23

7. Web Publishing Flowchart..25

8. Relationship between tables..27

9. Login page..28

10. Login2 page..29

11. Select page..30

12. Select Fertility test and most recent day's samples..34

13. Select Water & Salinity test and all samples..35

14. Select Forage test and date from _ to _ samples..36

15. Select Texture test and a specific sample number..37

16. Water & Salinity report page..40

17. Forage report page..41

18. Soil test interpretation for NO3-N..43

19. A Hardcopy report..50

20. A Previous Web report...51

21. A Web report for Fertility test...52

22. A Web report for Irrigation Water test..53

23. A Web report for Livestock Water test..54

24. A Web report for Household Water test...55

25. A Web report for Salinity Management test..56

26. A Web report for Comprehensive Salinity test.....................................57

27. A Web report with text box for county extension office.........................58

28. A Web report for Forage test...59

29. A Web report for Texture test..60

LIST OF TABLES

Table Page

1. Database tables used in this application...26

2. Reports request interpretations and recommendations...................................39

3. Crop Code and Yield Goal (YGandCode) table..42

I. INTRODUCTION

1.1 Background

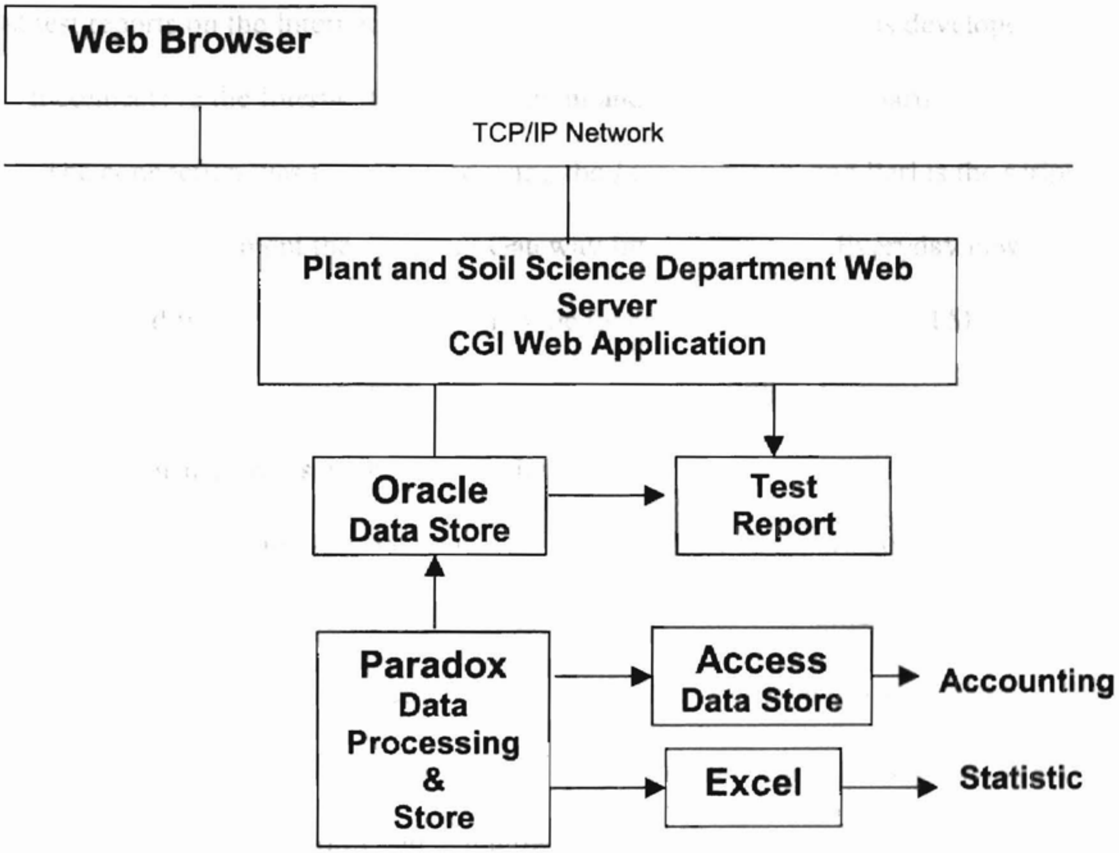
"In the past few years, the Internet has really made the world a small place; people from all over the world interact in ways not thought possible even a decade ago" [LeVitus, B. and Evans, J., 1996]. Through the Internet, information can be reached in an instant. Hundreds of thousands of businesses with something to say to the world are seeking affordable ways to enter this new electronic medium -- by publishing on the World Wide Web. "Launching a Web site is one of the most popular things to do on the Internet today" [Wehling, J., et al., 1996]. With the rapid progress of Internet based technology, people will rely more and more on the Internet to get information.

This thesis describes an application of "Dynamic Web Publishing" for the Soil, Water and Forage Analytical Laboratory (SWFAL) of OSU. The application is implemented with a relatively newly developed Microsoft technology – Active Server Page (ASP). This application establishes a dedicated Web server for the SWFAL information system, to provide its customers with the most updated dynamically generated testing reports and interpretations.

1.2 Problem Statement

The SWFAL was established by the Oklahoma Cooperative Extension Service in 1969. Every year over 30,000 samples are submitted for analysis by people throughout the state. The SWFAL provides valuable information that helps users to protect and utilize their soil, water and forage resources efficiently and effectively [McCray, B. J., 2000]. The SWFAL has already developed three database systems. Figure 1 is a simplified diagram of the current system structure.

Figure 1: The SWFAL current information system



The first management system was completed in 1994 using Borland Paradox 4.5. It is used as a local system. The daily testing results and calculated data are stored in this system, and all the hardcopy listings and reports are generated and printed from it. When statistical information is needed, selected raw data is copied to Microsoft Excel 97 to generate requested reports.

The second database system uses Microsoft Access. It is used to generate and store accounting information for all customers and vendors. The bills and invoices are printed from this system, and all its input data are transported from Paradox monthly.

When SWFAL decided to publish the testing results on the Web to its customers, they found it difficult to find someone who understood DOS based Paradox well enough to post test reports on the Internet. Consequently, an Oracle database was developed in 1999. It connects to the Internet through the Plant and Soil Sciences Department's Web server. The connection was implemented using the Java language, and Perl is the script language used to implement the Common Gateway Interface (CGI). Everyday new data must be uploaded from the Paradox system to the Oracle database, and an HTML report file is generated for each sample tested. The clients may view their testing results and print the reports using browsers through the Internet.

Traditionally computing and connectivity are performed separately. An application program creates information locally, then a connecting software such as a Web browser only displays the existing information. The web page is a static pure-HTML file; it is completely determined before the page is requested [Ullman, C., et al., 1999]. Using a new class of program -- internet-enabled programming, we can combine the business application and the communication together, so the client Web browser may have true computing power, and the database information may have true Internet connectivity [Simon, A. R. and Wheeler, T., 1998]. It enables the programmer to create a web page dynamically according to the web visitor's request.

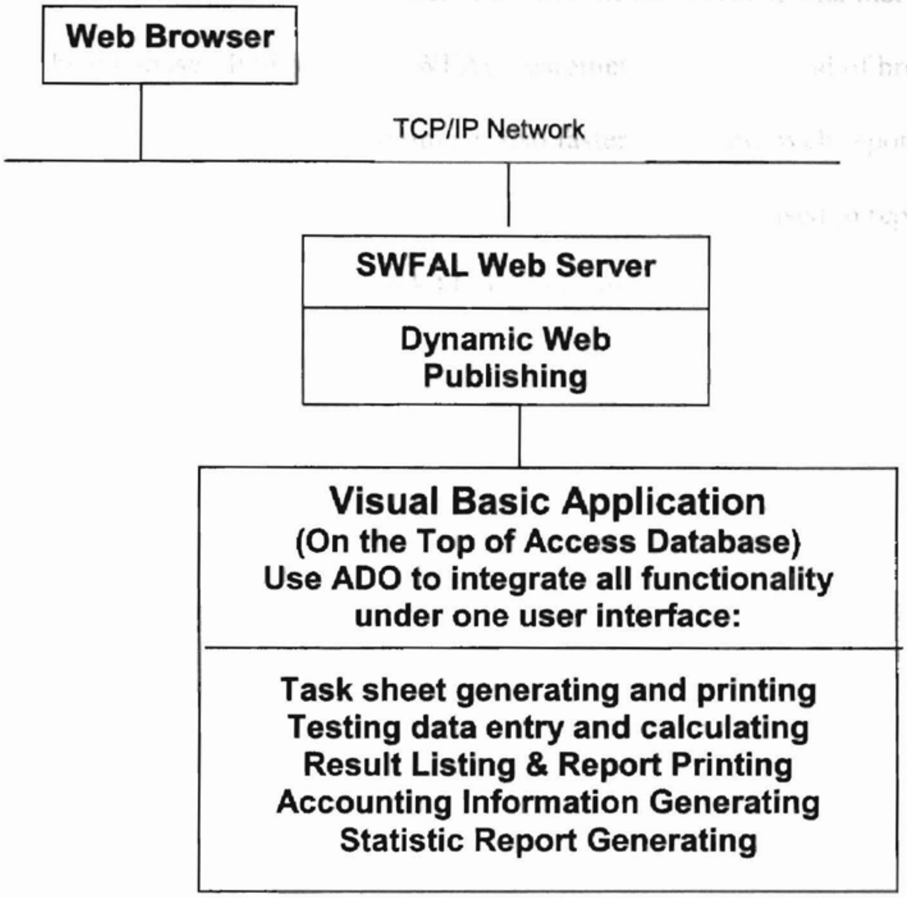
The current SWFAL web publishing is a combination of the dynamic and static models – it queries testing results for the authorized customer after he enters the correct customer ID and password. A customer can view only his/her own sample results, and this information is generated dynamically. For each testing sample there exists a static report file. It is stored in a directory at the server machine, and each sample's LabID is linked to its corresponding HTML report, so that when the customer clicks the selected sample, it will bring the report to the customer's browser.

With this current system, if any sample's data is changed, a new HTML report file must be regenerated to replace the original file, otherwise the data shown in the client browser may differ from the data stored in the database.

The current SWFAL information system is too difficult to manage and to expand due to the complexity of several different subsystems. For example, adding new testing items or services requires finding someone who has a good knowledge of all mentioned languages as well as software to modify the current system. After a few years of running this system, the laboratory decided to develop a new data management system that is more efficient and easy to operate. The goal of this new information system is to combine all the requested functions within a unified program, yielding a system that is easy to use, to maintain, and to provide the clients with direct database access through the Internet. Figure 2 depicts the new system design.

The new database operator or programmer deals with only one database system and does not require the knowledge of several different programming languages. The new system uses Microsoft Access as its database lower-layer, and uses Visual Basic to implement its data processing. The accounting information is generated directly

Figure 2. The SWFAL new information system


Web Browser
TCP/IP Network
SWFAL Web Server
Dynamic Web Publishing
Visual Basic Application
(On the Top of Access Database)
Use ADO to integrate all functionality under one user interface:
Task sheet generating and printing
Testing data entry and calculating
Result Listing & Report Printing
Accounting Information Generating
Statistic Report Generating

from the Access database. With Active Data Object (ADO) functionality, the program also embeds Excel applications to generate spreadsheets and statistical reports. All these functionalities can be implemented using one user interface.

1.3 The Objectives

The objects for this study is to establish a dedicated Web server for the SWFAL, and implement a complete dynamic Web publishing application to replace the previous CGI Web program. The new application improves the system performance and provides the SWFAL customers with better services. With this new application, a Web report is generated only upon customer's request. No test report is stored into the disk. It ensures that the data presented on the user's browser is always the most recent data that has been entered into the database. It allows the SWFAL customer to use any kind of browser to access their test report and the response time is also faster. The new Web reports are formatted similar to the hardcopy report, so the screen report can be used to replace the hardcopy report to save the time and SWFAL mailing cost.

II. LITERATURE REVIEW

2.1 Client – Server Computing:

Each computer program that communicates can be classified in one of two categories. Any program that offers a service belongs to the server category; any program that contacts a service belongs to the client category [Comer, D. E., 2000]. A server can provide services to multiple clients at the same time, and must keep running all the time, so clients may request services at anytime.

A client – server application is most often database-centric. And the implementations are based around Local Area Network (LAN). It has two essential characteristics [Marchal, B., 1999]:

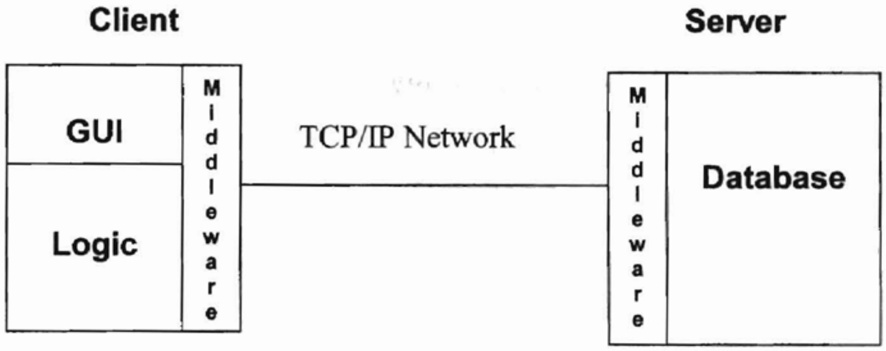
- They are distributed applications, meaning that two or more computers are connected over a network.
- The two computers have specific roles.

Two-tier Client-Server Application

The early client – server applications were usually implemented in two logic tiers. In most cases they were also separated into two separate hardware entities. This actually is the foundation of the client-server model.

A two-tier system means choosing tools for the client as well as the server. Both client and server have their own communication layers; this layer is called a middleware. The two middleware must be compatible with each other. The system does not function well if the client middleware is Open Database Connectivity (ODBC) and the database server does not support ODBC [Smith, N. E., 1998]. Figure 3 is a simple architecture of the two-tier client – server system.

Figure 3. A two-tier client-server model

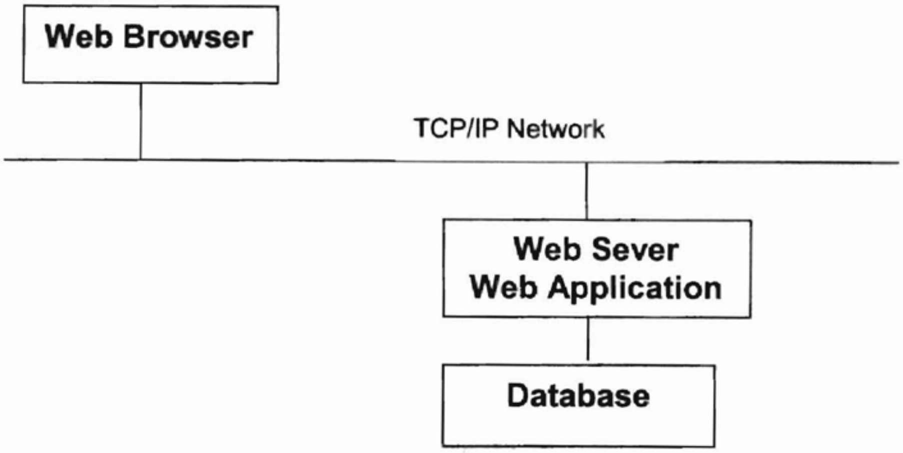


To overcome the limitations of a two-tier application, a three-tier model was developed. It removed the middleware portion from both client and server sides, and formed an independent middleware tier to communicate with the client and server.

Three-tier Client-Server Application

Currently most client – server applications use a three-tier architecture and build some Database Management System (DBMS) -like functionality in the middle-tier, often under-utilizing the DBMS [Nori, A. K., 2000]. Figure 4 is a three-tier client-server model.

Figure 4. A three-tier client-server model



A three-tier system adds the decision for implementing the third tier, which is an extra software layer on the server, where the business rules can be encoded. A third tier adds a second choice to make in the language category: the implementation language for the third tier [Smith, N. E., 1998]. This third tier is an independent middleware that sits between client and server, and provides communication between the client and the server.

Middleware

Middleware is the collection of distributed computing services that, within any given processing environment, enables client and server to communicate and interoperate with one another in the most expedient, flexible, and correct manner possible [Simon, A.R., and Wheeler, T., 1998]. Today, in most cases, the middleware plays a double role; it is a server when it communicates with a client, and it is a client when it communicates with a database server.

With this additional middleware layer, there are definite benefits to a three-tier client-server approach. These include:

- Scalability
- Generally lower network traffic
- Flexibility

2.2 Data Access:

Object Linking and Embedding Database (OLE DB)

OLE DB is a Component Object Model (COM) based, low-level programming interface. OLE DB is more powerful than ODBC in that it not only provides access to relational and non-relational database, it can also provide accesses to other types of data sources such as e-mail, voice, graphics etc.

OLE DB exposes two main components for connectivity to data: consumers and providers. Consumers, such as client-server or Web-based applications, use data. Providers are the components that talk to the data, translating the information and exposing a common interface for any type of consumer. The consumer does not need to know the data is being accessed; this ability is encapsulated within OLE DB itself. With OLE DB technology "Universal Data Access" (UDA) is no longer a dream.

ActiveX Data Object (ADO)

ADO is a programming interface from Microsoft that is designed as the Microsoft standard for data access. It was first used with Internet Information Server (IIS). ADO objects work across different platforms and programming languages, and support database accesses both for local as well as remote data objects (RDO).

ADO is a high-level interface, which is based on another low-level interface – OLE DB. "OLE DB objects are very low-level objects. Scripting languages like VBScrip (and even language like Visual Basis) are not sufficiently powerful to allow us to manipulate the OLE DB objects" [Ullman,C., et al., 1999]. ADO wraps OLE DB's functionality within its objects, makes the data access much easier at high level interface. ADO exposes OLE DB's capabilities while hiding its complexities. So using ADO, the data access actually works through two layers. There are only four objects and two collections in the ADO object model:

1. Connection Object: It provides the link to the data source.
2. Command Object: It allows the programmer to make queries from the data store.
   Parameters Collection (depends on Command Object)
3. Recordset Object: The most important object for data access. With its properties, methods and events, data manipulation can be easily achieved.
   Fields Collection (exists within the Recordset Object)
4. Errors Object: It is created when an error occurs while accessing the data, to provide a powerful debugging tool.

ADO can be used in various applications. To develop Web application, ADO is programmable via VBScript and JavaScript, and also supports specialized abilities that make it more Web-friendly – and is therefore likely to be the best choice for data-driven Web applications [Para, J., et al., 1999].

2.3 Markup Language:

Markup language deals with everything relating to the appearance, rather than the content of a document.

Standard General Markup Language (SGML)

Standard General Markup Language (SGML) was created in 1986 from a preexisting prototype called General Markup Language (GML) by IBM. SGML is one of the most generally accepted markup languages because it essentially deals with organizing information. Before SGML, the sole function of a markup language was establishing a quick method of consistent formatting of a document. Formatting was placed above content. SGML has taken a different approach; it concentrates primarily on content with a secondary emphasis on formatting. SGML tags text depending on the information within the text [Townsend, J. J., et al., 1996].

"Using Document Type Definition (DTD), SGML approaches the maximum amount of standardization that can be achieved in document design without sacrificing flexibility" [Matzen, R. W., et al, 1997]. SGML makes standardized documents which can be generated much more efficiently, especially when in a large quantity.

HyperText Markup Language (HTML)

HTML is a subset of SGML. It was used originally as a way for physicists to format shared electronic documents at CERN (Conseil Europeen pour la Recheche Nucleaire) in Geneva, Switzerland [Smith, N. E., 1998].

HTML is a markup language that can be interpreted by all World Wide Web (WWW) browsers. An HTML document is a plain text file that includes additional markup tags. These tags control the way that the document's text is formatted, which pictures are displayed, where links point, etc. [Aitken, P., 1996].

The first version of HTML had a dozen tags [Marchal, B., 1999]. The latest HTML specification has well over a hundred [Smith, N. E., 1998]. As the HTML can present more and more variety of media contents, it also becomes more and more complex. And still, the clients want HTML to provide more abilities. This raises the need for a new generation of markup language, that is more flexible and powerful, yet simpler.

eXtensible Markup Language (XML)

The eXtensible Markup Language (XML) is being standardized by the World Wide Web Consortium (W3C). It addresses the need for semantic markup but not the requirement that publishers agree on a common set of classes for semantic markup to be useful.

XML exists because HTML was successful. XML incorporates many successful features of HTML. XML also exists because HTML could not meet the new demands. XML is extensible because it predefines no tags but lets the author create tags that are needed for his or her application [Marchal, B., 1999]. It promises to return the Web to the content-based structure instead of the format-based structure. It allows developers to create a markup structure on logical content rather than formatting [Ramakrishnan, R., 2000].

XML is unlikely to replace HTML in the near or medium-term. Work is already under way to combine XML and HTML in XHTML, an XML version of HTML [Marchal, B., 1999].

2.4 Dynamic Web Page:

Web pages can be catagorized in static Web pages and dynamic Web pages. Static Web pages also can be called a "materialized WebView" and dynamic Web pages can be called a "virtual WebView".

Originally, the Web was designed for accessing static files, which were stored in the database or Web server. When a Web page is requested, the same content will be shown on the browser, no matter who requested it or when it is requested. With the development of Internet technologies, the Web page now can be computed dynamically on-demand. In the static case, since the Web page is pre-computed, it takes less time for the server to service the access request. But every update to the data stored in database leads to an update to the Web page, increasing the server load. On the other hand, in the dynamic case, the cost to compute the Web page increases the service time, but the Web page always presents the most recent data.

In general, when a Web page is shared by multiple viewers, "it should be considered for materialization in order to improve the Web server's performance". If the web page results from an arbitrary querie and it is not shared with other viewers, it is not need to be stored [Labrinidis A., Roussopoulos N.].

To create dynamic Web applications Common Gateway Interface (CGI) and Active Server Page(ASP) are the most popular technologies.

Common Gateway Interface (CGI)

CGI is a program written in almost any program language that the Web server supports. It functions like the glue between HTML pages and other programs. CGI is a mechanism for creating scripts on the server. These scripts then can be used to create dynamic Web applications [Ullman, C., et al., 1999].

CGI allows a Web server to associate a given Uniform Resource Locator (URL) with an external computer program instead of a static document on a disk. When a browser requests one of the special URLs, the server does not fetch a page from a file on a disk; instead, the server runs the associated computer program and sends the output from the program back to the user. A server can have an arbitrary number of CGI programs that perform different computations; the server uses the URL in the incoming request to determine which CGI program to run. The interaction between a Web server and a CGI program depends on the computer's operating system [Comer, D. E., 2000].

Although we can generate dynamic Web pages using CGI technology, the CGI script cannot communicate directly with a user. The interaction between the CGI program and the user is realized via another Web technology – "Form Technology". The browser provides the form for the user to send his personal information or requests as the parameters; the roll of CGI is to pass the user-supplied date to the external program for processing.

Each time that a request arrives, the Web server starts processing anew. If there are many clients who access the same application at same time, and if each one requests several pages, then the performance can be very slow. Also, the most common programming languages used for CGI are C, C++, and Perl; none of them are easy languages to learn [Ullman, C., et al., 1999].

Active Server Page (ASP)

ASP is a Microsoft technology that is used to generate dynamic Web pages. It is designed to execute on the Microsoft Internet Information Server (IIS).

An ASP is a way of combining scripting code (which executes on the server as the page is loaded by the Web server) with HTML and other Web page content [Homer, A., 1998]. When asked to deliver an ASP, the Web server retrieves it from the file system, and arranges it to be handled by the ASP script host. The ASP script host is present on the Web server machine in the form of the file asp.dll, which is run by the Web server itself. After the script code is processed, the results are embedded into the HTML code and returned to the visitor's browser. Coding in ASP enables us to ensure that things like sensitive database queries and proprietary formulas are kept away from prying eyes.

The asp.dll file is compiled as a Windows dynamic link library (DLL). It means that the programmer can access the Web server's functions directly via the asp.dll, just by writing ASP code into the Web pages [Ullman, C., et al., 1999].

ASP encapsulates the properties and methods of the following server built-in objects:

ASP encapsulates the properties and methods of the following server built-in objects:

- Application: the sum of all files that can be accessed through a given virtual directory. It is same for all clients that use the application. When a client requests the URL, a copy of the application begins to run. The application level variable is shared by all clients, so lock and unlock is needed when it is accessed.

- ASPError: handles the errors that occurred in the script.
- ObjectContext: guarantees that a segment of code that is processed succeed completely or fail as a group.
- Request: obtains information from the client. It has five main methods – QueryString, Form, Cookies, ServerVariables, clientCertificate
- Response: sends information to the client or transfer to another page. It has two main methods – response.write and response.redirect.
- Server: represents the server itself, and provides common functions to create new objects.
- Session: represents a user and keeps the individual user's information between multiple pages during his session time. Session level variables are defined in Global.asa, which is a default file shared by all users.

These objects are part of the asp.dll and are available to ASP applications [Ullman, C., et al., 1999, Wang, G. H. and Shu, Y. T., 1999, Weissinger, A. K., 2000].

III. APPLICATION REQUIREMENTS AND DEVELOPMENT TOOLS

3.1 Business Requirements Analysis

There are several ways to implement most Web applications; the choice one makes has a large impact on performance [Smith, N. E., 1998]. In order to make a good choice, we first list all the business characteristics and the corresponding Web application requirements:

1. The SWFAL yearly sample processing is about 40,000. Of course, this number may increase in the future, but probably not very rapidly. This is not a large amount from the point of view of computer processing.
2. There will probably be fewer than 30 customers accessing the SWFAL Web site simultaneously, so network traffic will be light.
3. The testing results stored in the SWFAL database can be viewed only by authorized customers through Web browsers, and no customer is allowed to modify, insert or delete any data.
4. Authorized customers should be able to access the SWFAL Web site using any computer with any browser.
5. Currently there are four main testing types and eight report formats, the interpretations of the testing results requires substantial logic processing.
6. Information security

3.2 Selection of Implement Tools

In addition to the above factors, the server platform and related software also play an important role in making the decision. The SWFAL uses Windows 2000 as the new operating system, and the installed Internet Information Server (IIS) 5.0 is used as the Web server. As mentioned before, the new data processing application has been implemented with Visual Basic 6.0 and Microsoft Access 7.0. The selected Web development tool should be very compatible with the Web server and other software.

To create a dynamic Web page, a scripting language must be used. First, client-side script or server-side script -- which one to choose? Below is a brief review of the advantages and disadvantages of choosing a client-side script.

Advantages:

1. Response time is often faster than the server-side script, because the interpretation is processed at the client computer. It does not need many network trips.
2. Web server's workload is reduced, so more clients can access the Web site at the same time.

Disadvantages:

1. Client-side script is browser specific, because some browsers do not have the capabilities of interpreting certain script languages.
2. The code in client-side script is visible to the user. If one wants to hide these codes from the browser, then additional complex software will be needed.

It is obvious that the advantages of the client-side script will not bring much benefit to the application, because the workload at the server side is not large, and the network traffic is low. But the disadvantages are surely what we want to avoid, especially since we must make sure all customers will be able to access their sample results through the Web, no matter what computer and what browser they use. So the client - side script is not the choice.

Secondly, which scripting language to use? In our case, this seems relatively easy to decide, since the information system is implemented with Visual Basic, so the VBScript is the natural choice. VBScript is a subset of Visual Basic, and is the default script language of Windows 20000 IIS 5.0. To create dynamic Web information for SWFAL, the Active Server Page (ASP) 3.0, which uses VBScript as its scripting language, is a good development tool.

As mentioned before, the SWFAL current Web application is implemented with CGI mechanism. Although CGI is easy to use, it is not a high-performance programming interface. With CGI, each request received by the server, a new processing environment must be built, and each environment initialization requests processing overhead. Spawning new processes consumes server RAM and wastes the server's application resources [Townsend, J. J., et al., 1996].

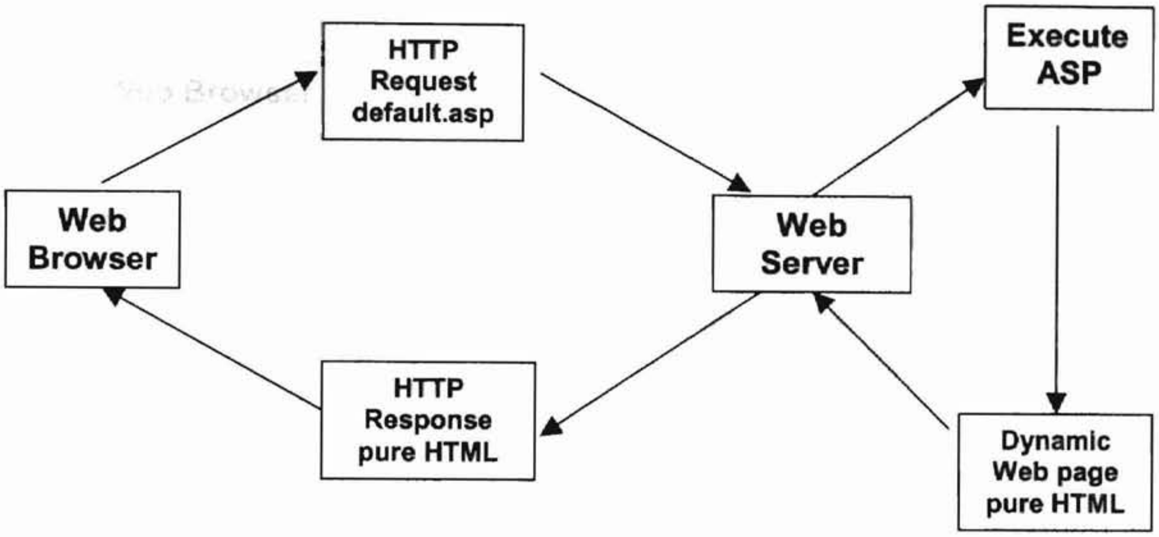
With ASP technology, when a client establishes a connection with the server, a user session is started, and it continues until the client leaves the Web site or the session's timeout is reached. During this period, the parameter values that the client entered can be stored in the application level variable(s) or session level variable(s), and these variables can be used by all pages within the session. It saves processing resources and greatly improves the performance of the system.

ASP is not a language, but it does make use of existing scripting languages such as VBScript or JavaScript. It can be written with any text editor or web page editor. An ASP file is likely to be composed of a combination of three types of syntax – some script code, some HTML tags and some pure text [Ullman, C., et al., 1999]. It even allows different scripts such as VBScript, JScript (a script compatible with JavaScript) or other script languages to be used within the same file. The ASP file must saved with an .asp extension and stored under an executable directory on the server computer.

On the client side, a Web browser is needed for a Web-based application. Today over 90 percent of the Web browser market is dominated by Netscape Navigator and Microsoft Internet Explorer [Simon, A. R. and Wheeler, T., 1998]. While they are more or less equal, Navigator and Explorer are not always compatible, especially when used as clients in a client-server environment [Smith, N. E., 1998]. Although we may assume most SWFAL customers will use Netscape Navigator or Internet Explorer, there may still be some customers who will use other browsers. Compatibility is a concern.

Unlike some other dynamic web applications, development tools such as Java applets, dynamic HTML, ActiveX controls, all of which are browser specific, the power of ASP lies in two facts: the HTML is not created until the user wants to see the web page, and second, it doesn't care what web browser is being used. A Web visitor can view ASP driven web pages from any computer, and with any modern browser. The ASP code is processed on a web server to generate the HTML file, then the HTML file is sent to the client and processed solely on the browser. It is faster than client-side dynamic web technologies such as Java applets or ActiveX controls, because in the end, the Web visitor only downloads a page of HTML. In addition, ASP provides improved security measures, since one can write code that cannot be viewed by end users. [Ullman, C., et al., 1999]. After it has been processed and sent back to the browser, the source code that the client may view is only pure HTML.

Figure 5. Web publishing with ASP technology

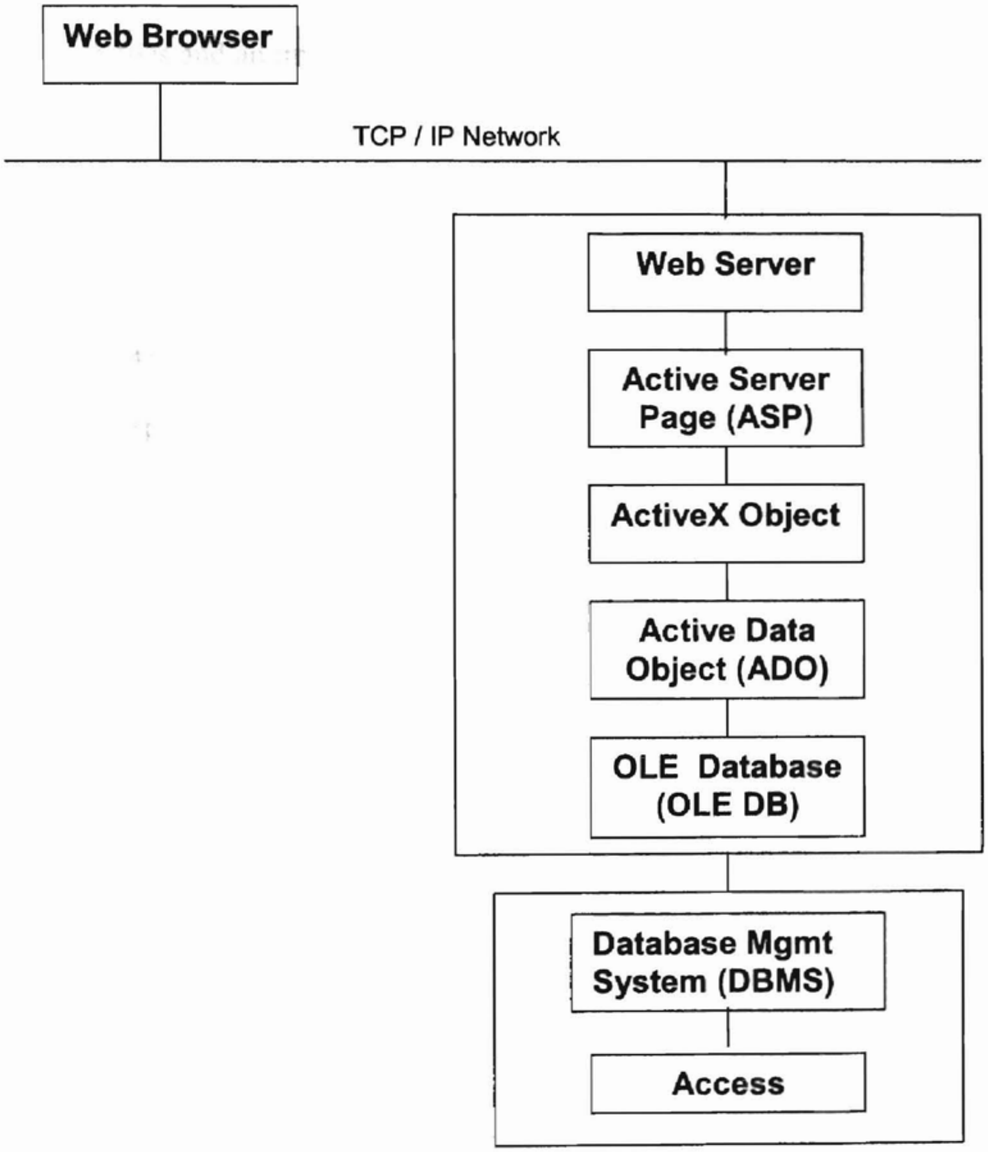


To implement this application, additional middleware is added to the Web server. The client portion handles the user's request. The server portion responds to the client's request. A middleware portion, an extra software layer on the server side that containing a mixture of HTML and Web Scripting code, works through several layers of software, opens the database, runs the queries, processes the logic and formats the results. The Web server then transmits the results to the Web visitor's browser [Viescas, J., 1999]. Figure 6 is the new structure of the three-tier Client/Server Web server.

Since Visual Basic, Access, and ASP (which uses the VBScript language) are all within the same Microsoft software family and closely related, the use of this system is easier to learn than it would be with software elements not designed to be used together.

Also, it is easier for the future programmer to modify or to add new functions. All of these advantages of using ASP technology are well suited for the application's purpose.

Figure 6. A three-tier client/server architecture implemented with ASP



IV. IMPLEMENTATION

4.1 General:

As mentioned before, this application is run by the Internet Information Server (IIS 5.0), which is built-in within the Windows 2000 operating system. This application includes 25 files and an image folder to keep .jpg files.

global.asa
Verify.asp
DataStore.asp
RecToTable.asp
ReportTitle.asp
InterSoil.asp
Water1.asp
Water2.asp
Water3.asp
Salinity5.asp
Salinity6.asp
ForageReport.asp
Visitorcount.asp

Login.asp
Login2.asp
Select.asp
ReportHead.asp
FertilityReport.asp
SalinityReport.asp
Inter1.asp
Inter2.asp
Inter3.asp
Inter5.asp
Inter6.asp
TextureReport.asp

All these files must be stored under the default Web directory "wwwroot" and must have the application property to be executable. Figure 7 is the flowchart of this Web application (the middleware portion). The information needed for this application

Figure 7: Web Publishing Flowchart

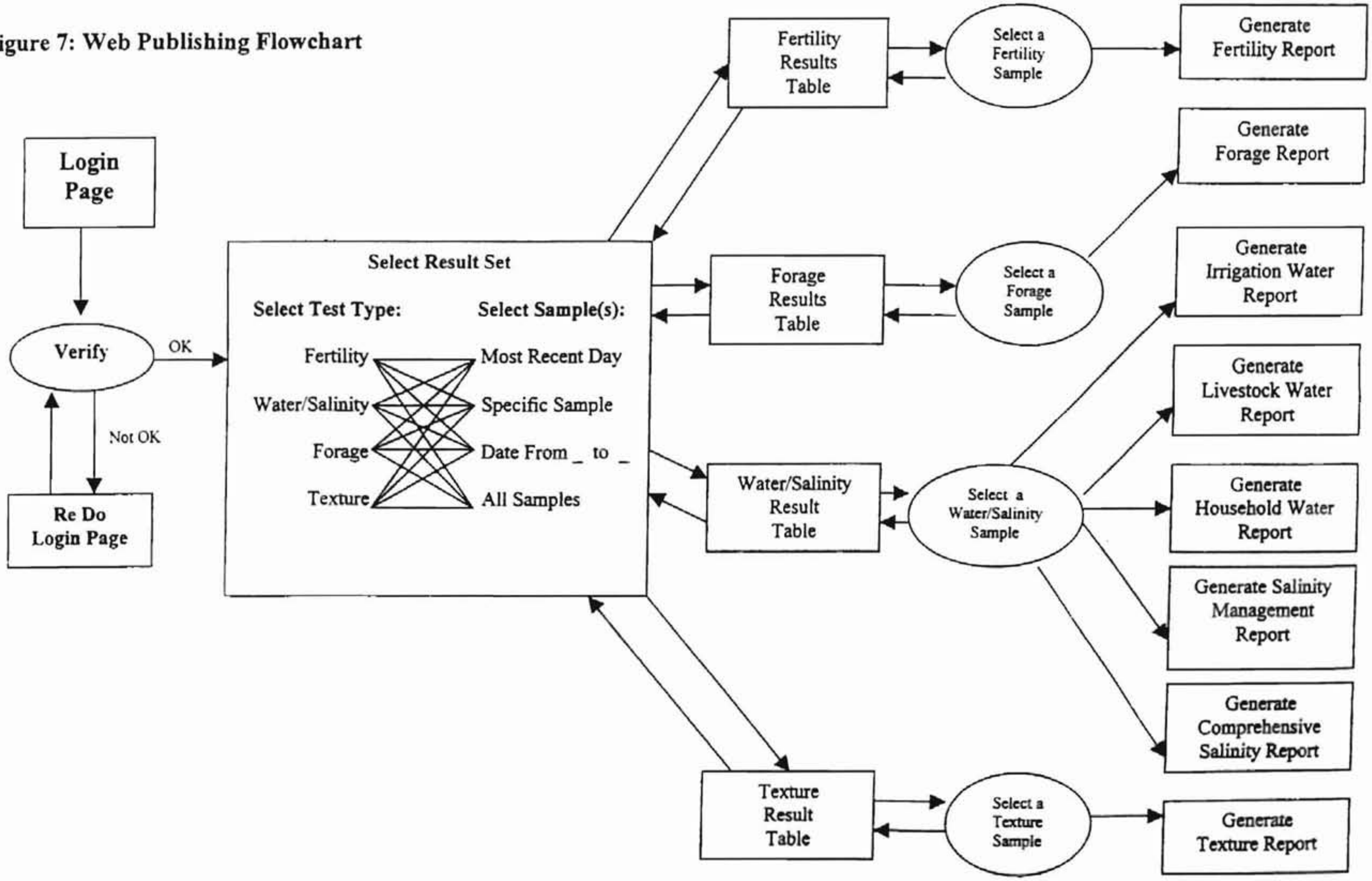
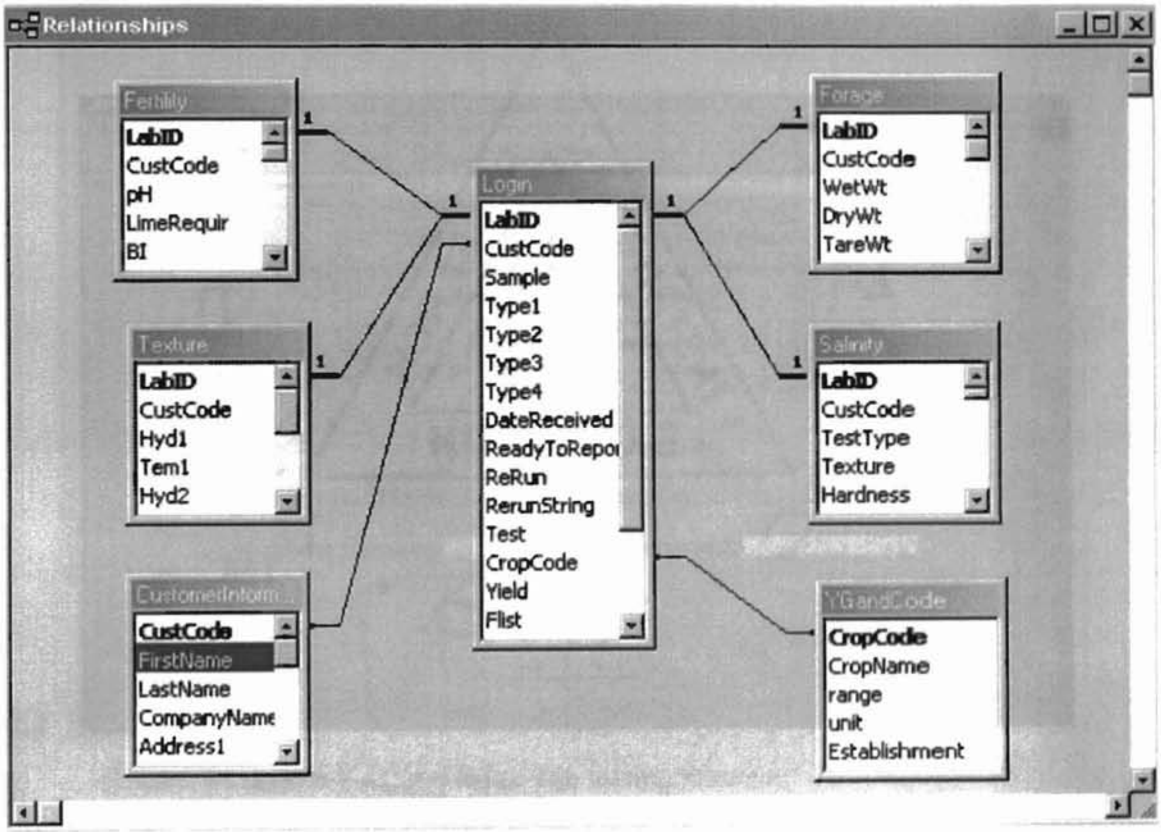


Table 1: Database tables used in this application

Table Name	Attributes
Login	LabID, CustCode, Sample, Type1, Type2, Type3, Type4, DateReceived, Rerun, Test, ReadyToReport, CropCode, Yield, Flist, ReportPrinted, Invoice, DateBilled
Customer Information	CustCode, FirstName, LastName, CompanyName, Address1, Address2, City, State, Zip, AccoutNum, Contact, homeNumber, workNumber, FaxNumber, Emailaddress, PriceCode, type, Status, Password
YGandCode (Crop Code)	CropCode, CropName, range, unit, Establishment
Fertility	LabID, CustCode, pH, LimeRequir, BI, Surno3, SubNo3, T_No3, NRequir, P, PRrequir, PPercent, K, KRequir, KPercent, SurSo4, SubSo4, T_So4, SRequir, Ca, CaRequir, Mg, MgRequire, Fe, FeRequire, Zn, ZnRequire, B, BRequire, Type
Salinity	LabID, CustCode, TestType, Texture, Hardness, EC, ECC, pH, Na, Sodium, %Na, Ca, Calcium, Mg, Magnesium K, Potassium, Cl, chloride, NO3N, SO4, Sulfate, CO3, Carbonate, HCO3, Bicarbonate, Boron, TSS, PAR, SAR, EPP, ESP, RSC, SUM CATION, SUM ANIONS, SumEC, NH3-N, NH3_det, Ortho-P, Ort_det, IDAP-P, Other, Alkalinity, NoInter
Forage	LabID, CustCode, WetW, DryWt, TareWt, DryInt, DryFinal, Nitrate, Protein, ADF, NDF, CorNitrate, CorProtein, CorADF, Cor NDF, Moisture, CalRFV, CalProtein, DalTDN Maint, Lact, Gain
Texture	LabID, CustCode, Hyd1,Tem1, Hyd2, Tem2, Weight, Texture, Sand, Clay, Silt

comes from the seven database tables shown on Table 1. The relationship among these tables is illustrated on figure 8.

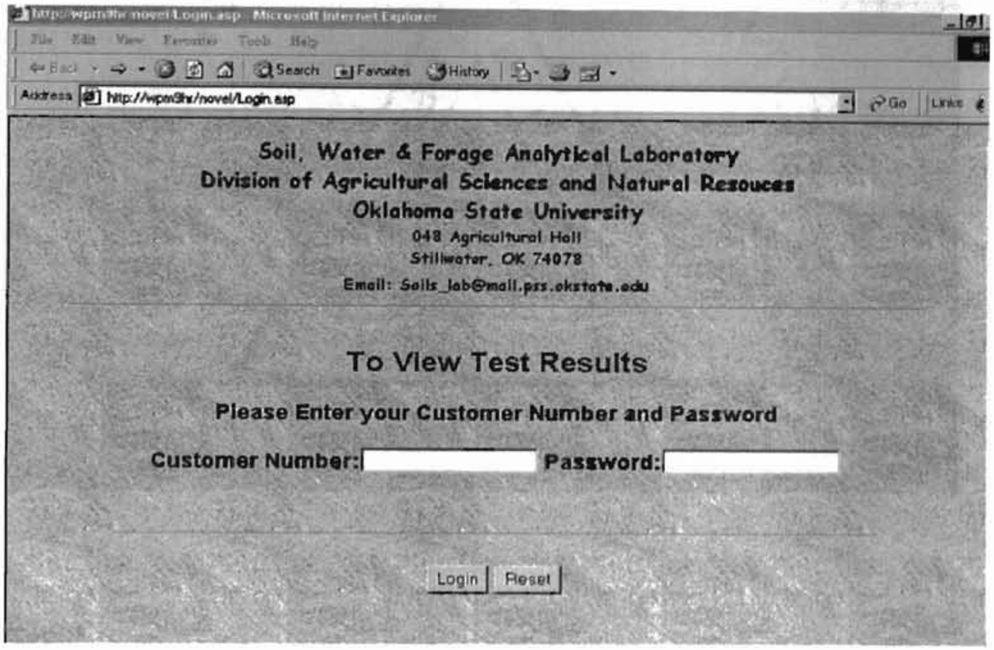
Figure 8: Relationship between the tables



4.2 Login:

The Login.asp presents a form for the user to enter his customer number and password, then passes the information to Verify.asp function. Figure 9 presents the Login interface.

Figure 9: Login page



http://wpm9hr/novel/Login.asp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://wpm9hr/novel/Login.asp
Go Links

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall
Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

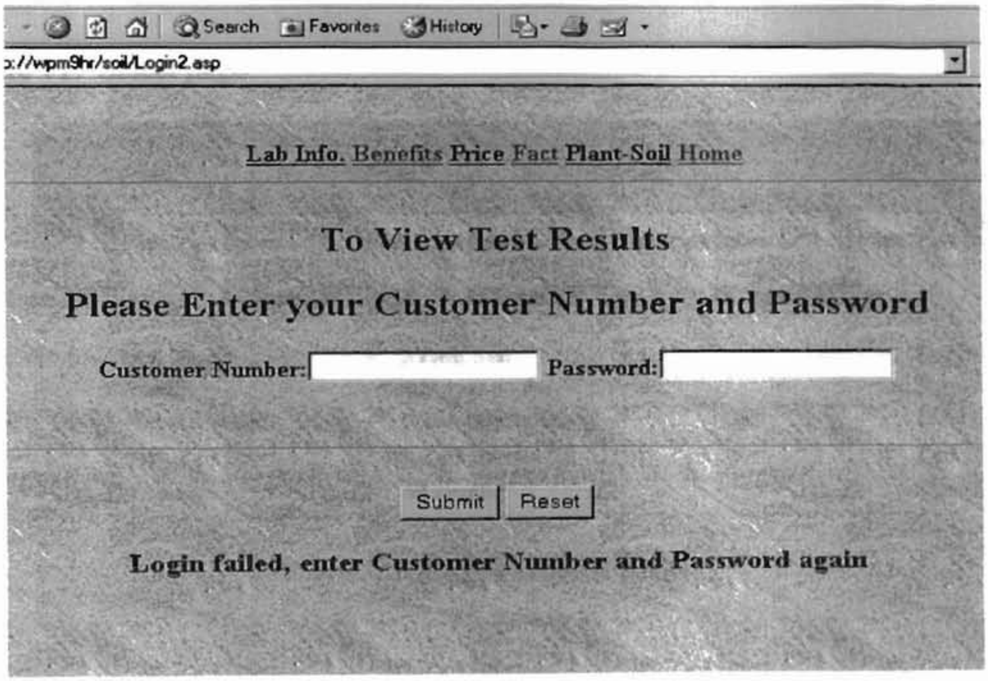
To View Test Results

Please Enter your Customer Number and Password

Customer Number: Password:

The Verify.asp queries the password data from the CustomerInformation table then compares the input information with the stored password, if it matches the customer's password, this function redirects the user to the Select.asp page and stores the customer number into a session level variable. This variable is used as the criterion to query the information for this particular customer only. If the input cannot match the stored password, the user is redirected to the Login2.asp page, which asks the user to redo the longin. Figure 10 presents the Login2 page.

Figure 10: Login2 page


Search
Favorites
History
p://wpm9hr/soil/Login2.asp
Lab Info. Benefits Price Fact Plant-Soil Home

To View Test Results

Please Enter your Customer Number and Password

Customer Number: Password:

Login failed, enter Customer Number and Password again

4.3 Select

Currently, the SWFAL provides four testing services:

Soil

Water & Salinity

Forage

Texture

For each test type, four ranges of sample set can be selected:

Most Recent Day

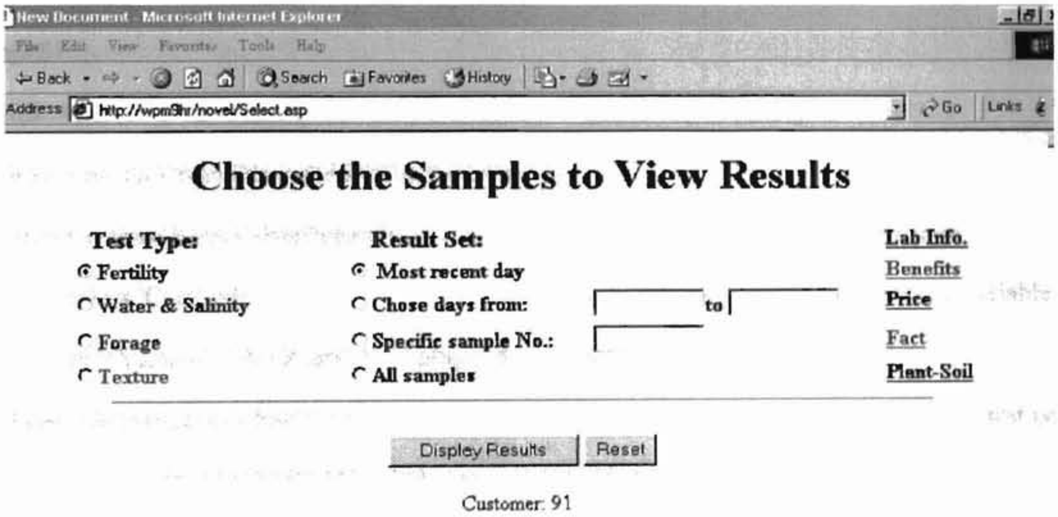
Specific Sample Number

Date from __ to __

All Samples

There are total of 16 choices that a client can make. When a user chooses a test type, a session level variable stores this information. (This is necessary because there are possibilities that a single sample may request two or three different tests). Later, it becomes known from which table to get the results. Figure 11 is the Select page interface.

Figure 11: Select page


New Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://wpm9hr/novel/Select.asp
Go Links

Choose the Samples to View Results

Test Type:
● Fertility
○ Water & Salinity
○ Forage
○ Texture

Result Set:
● Most recent day
○ Chose days from: [ ] to [ ]
○ Specific sample No.: [ ]
○ All samples

Lab Info.
Benefits
Price
Fact
Plant-Soil
Displey Results
Reset

Customer: 91

Following is part of the code of the "Select.asp" function. It illustrates the selected test type for Fertility, and how to generate the four resulting tables. The codes for the remaining three test types are similar, but use different tables and attributes.

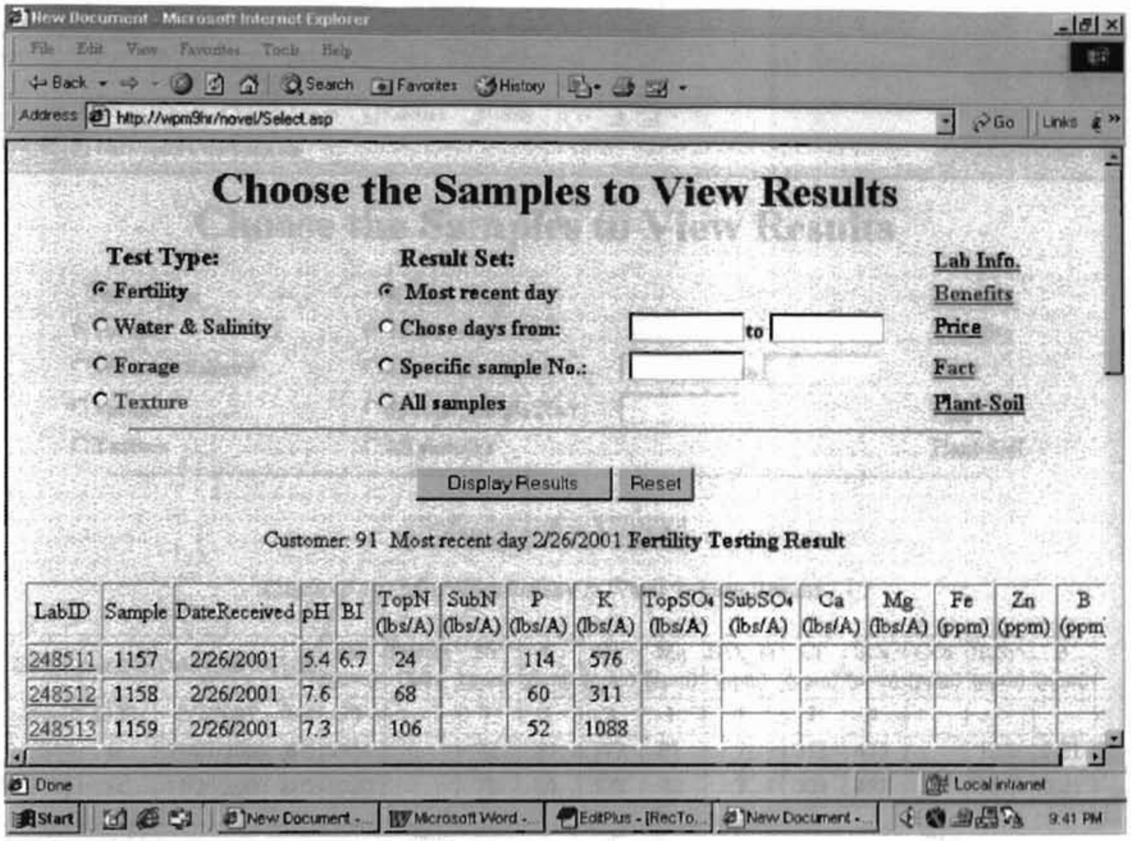
```
<%

 dim dataOption                                    ' selected sample range

 dim strRecentDay                                  ' most recent day

 dim strStartDay                                   ' begining date

 dim strEndDay                                     ' end date
dim strSample                                              ' specific sample

dim strTestType                                            ' selected test type

dim Criteria                                               ' query key

dim Criteria2

dim strSQL                                                 ' SQL query string

dim strSQL1

dim strSQL2

dim objRS                                                  ' record set object

dim objRS1

set objRS  = Server.CreateObject("ADODB.Recordset")   'define record set object

set objRS1 = Server.CreateObject("ADODB.Recordset")

dataOption = request.Form ("dataOption")                  'get selected range from input form

Criteria = Session("CustCode")                            'get customer number from session variable

response.write "Customer: " & Criteria & "&nbsp;" & "&nbsp;"

strTestType = Request.Form("testType")                    'open recordset according to selected test type

'*******************************************************

'If Test type choose Fertility

if strTestType = "Fertility" then

  Session("SelectedType") = "Fertility"                   'store seleceted test type

                                                          'query record set

 strSQL = "select  Login.LabID,Sample,DateReceived,pH,BI,Surno3," & _

          "SubNo3,P,K,SurSo4,SubSo4,Ca,Mg,Fe,Zn,B " & _

          "from  Fertility, Login " & _

          "where  Login.CustCode = " &  Criteria & _

          " AND Login.LabID = Fertility.LabID AND ReadyToReport = True  order by DateReceived

           DESC,Sample ASC"                               ' in date descent, sample ascent order

 objRS.Open strSQL, strConnect, adOpenStatic, adLockReadOnly

 If Not(objRS.BOF and objRS.EOF) then                     'if record set is not empty
```

```
Select Case dataOption
  Case "0"                                                'for all samples
    response.write "<b>All of Your <font face color=#CC0000>Fertility </font>Testing
                   Results</b><BR><BR>"
    response.write RecToTable(objRS)
  Case "1"                                                'for all most recent day samples
    objRS.MoveFirst                                       'the first record is the latest sample since
                                                          'they are in descent date received order
    response.write "Most recent day "
    strRecentDay = objRS.Fields("DateReceived")           'get the login date from this record
    response.write strRecentDay
    response.write "<b> <font face color=#CC0000>Fertility</font> Testing
                   Result</b><BR><BR>"
    objRS.Filter = "DateReceived = " & strRecentDay       'filtering all the records of this day
    response.write RecToTable(objRS)                      'print a table
    objRS.Filter = adFilterNone                           'clear the filter
  Case "2"                                                'select a specific fertility sample
    strSample = CStr(request.Form("sampleNumber"))        'direct query is fast than filter
    response.write "Sample No.: " & strSample & "&nbsp;&nbsp;" & "<b><font face
    color=#CC0000>Fertility </font>Testing Result</b><BR><BR>"
    Criteria2 = "select  Login.LabID,Sample,DateReceived,pH,BI,Surno3," & _
                "SubNo3,P,K,SurSo4,SubSo4,Ca,Mg,Fe,Zn,B " & _
                "from  Fertility, Login " & _
                "where Login.CustCode = " & Criteria & " AND Login.LabID = Fertility.LabID " &
                " AND ReadyToReport = True and Sample='" & strSample & "'"
    objRS1.open  Criteria2, strConnect, adOpenStatic, adLockReadOnly
    if (objRS1.EOF and objRS1.BOF) then                   'if no sample found
       response.write strSample & " no record find "
```

```
      else
        response.write objRS1("Sample")
        response.write RecToTable(objRS1)
      end if
      objRS1.Close
    Case "3"                                     'get samples from startDay to endDay
      strStartDay = request.Form("startDay")     'get input start day
      strEndDay = request.Form("EndDay")         'get input end day
      response.write "From: " & strStartDay & "&nbsp;" & "To : " & strEndDay & "&nbsp;" &
      "<b><font face color=#CC0000>Fertility</font> Testing Results</b><BR><BR>"
                                                 'filtering records between these dates
      objRS.Filter = "DateReceived >= " & strStartDay  & " and DateReceived <= " & strEndDay
      response.write RecToTable(objRS)           'print result table
      objRS.Filter = adFilterNone
  end Select
Else
  response.write "<b>No Fertility Sample Found</b>"
End if
objRS.Close
'***********************************************
```

The Select.asp page takes the selected test type and sample range, queries the required record set from relative database tables, then calls the "RecToTable" function to present the record set table. Figure12 through Figure 15 present four combinations of different results.

Figure 12: Select Fertility test and most recent day's sample


New Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://wpm9hr/novel/Select.asp
Go Links
Lab Info.
Benefits
Price
Fact
Plant-Soil
Display Results
Reset
Done
Local intranet
Start New Document - Microsoft Word - EditPlus - [RecTo... New Document - 9:41 PM

Choose the Samples to View Results

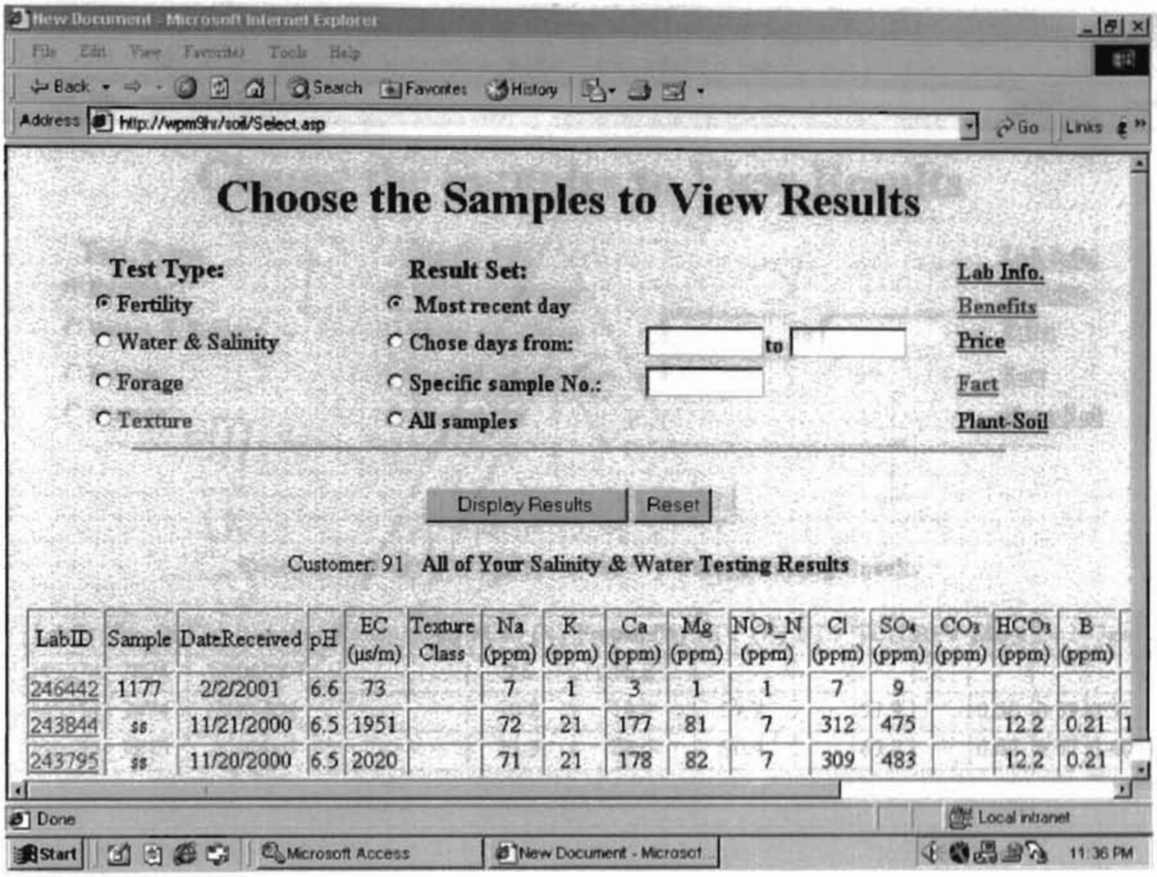
Test Type:
- ● Fertility
- ○ Water & Salinity
- ○ Forage
- ○ Texture

Result Set:
- ● Most recent day
- ○ Chose days from: ____ to ____
- ○ Specific sample No.: ____
- ○ All samples

Customer: 91 Most recent day 2/26/2001 **Fertility Testing Result**

LabID	Sample	DateReceived	pH	BI	TopN (lbs/A)	SubN (lbs/A)	P (lbs/A)	K (lbs/A)	TopSO4 (lbs/A)	SubSO4 (lbs/A)	Ca (lbs/A)	Mg (lbs/A)	Fe (ppm)	Zn (ppm)	B (ppm
248511	1157	2/26/2001	5.4	6.7	24		114	576							
248512	1158	2/26/2001	7.6		68		60	311							
248513	1159	2/26/2001	7.3		106		52	1088							

Figure 13 : Select Water & Salinity and all samples


New Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://wpm9hr/soil/Select.asp
Go Links

Choose the Samples to View Results

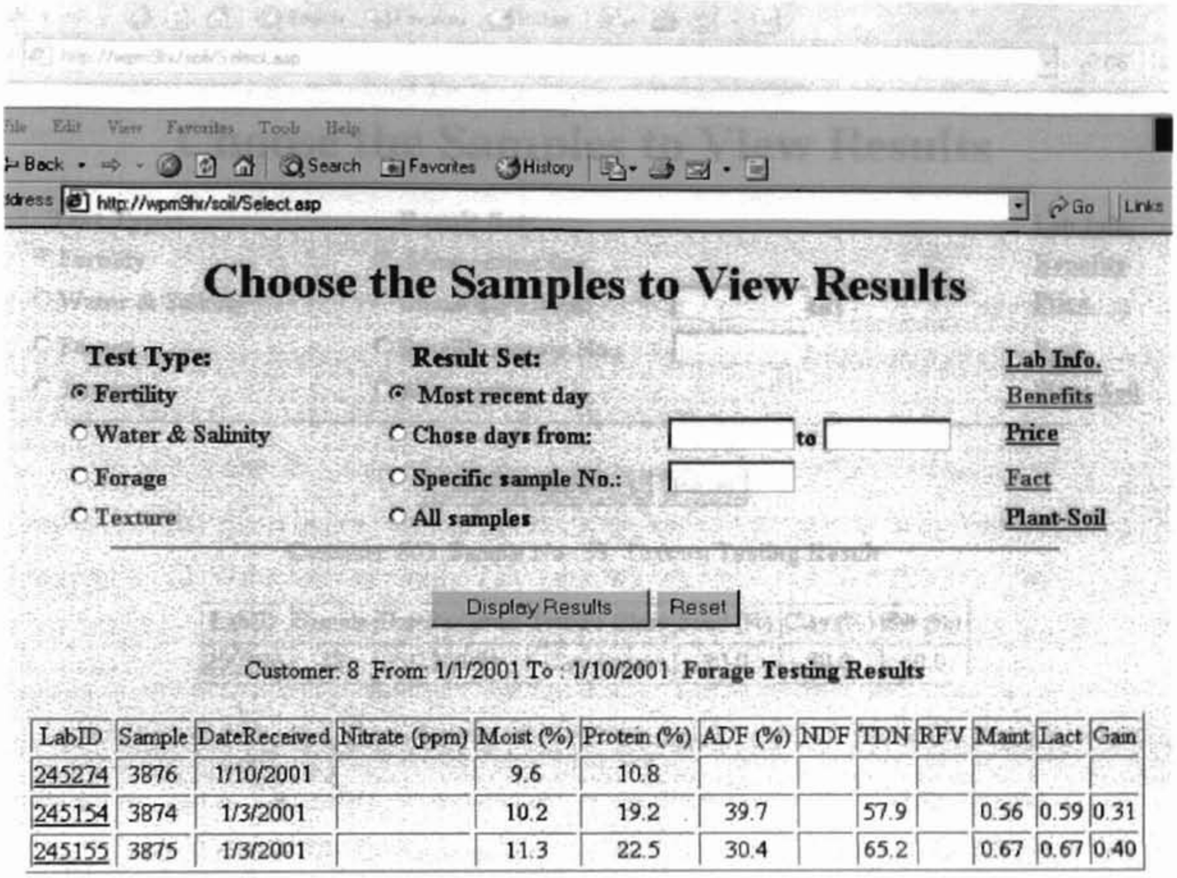
Test Type:	Result Set:			
● Fertility	● Most recent day			Lab Info.
○ Water & Salinity	○ Chose days from:	to		Benefits
○ Forage	○ Specific sample No.:			Price
				Fact
○ Texture	○ All samples			Plant-Soil

Customer: 91 **All of Your Salinity & Water Testing Results**

LabID	Sample	DateReceived	pH	EC (µs/m)	Texture Class	Na (ppm)	K (ppm)	Ca (ppm)	Mg (ppm)	NO_3_N (ppm)	Cl (ppm)	SO_4 (ppm)	CO_3 (ppm)	HCO_3 (ppm)	B (ppm)	
246442	1177	2/2/2001	6.6	73		7	1	3	1	1	7	9				
243844	ss	11/21/2000	6.5	1951		72	21	177	81	7	312	475		12.2	0.21	1
243795	ss	11/20/2000	6.5	2020		71	21	178	82	7	309	483		12.2	0.21	

Figure 15 : Select Texture test and a specific sample number

Figure 14 : Select Forage test and date from _ to _ samples



File Edit View Favorites Tools Help
Back Search Favorites History
Address http://wpm9hr/soil/Select.asp
Go Links

Choose the Samples to View Results

Test Type:

- ● Fertility
- ○ Water & Salinity
- ○ Forage
- ○ Texture

Result Set:

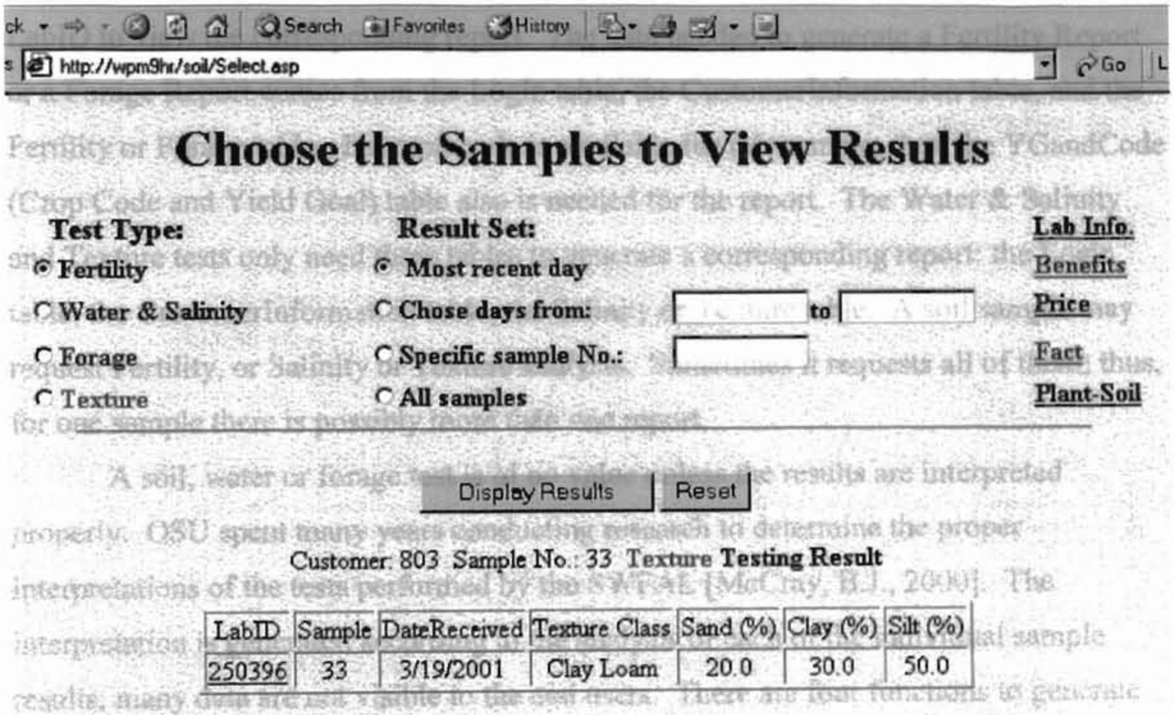
- ● Most recent day
- ○ Chose days from: ___ to ___
- ○ Specific sample No.: ___
- ○ All samples

Lab Info.
Benefits
Price
Fact
Plant-Soil

Customer: 8 From: 1/1/2001 To : 1/10/2001 **Forage Testing Results**

LabID	Sample	DateReceived	Nitrate (ppm)	Moist (%)	Protein (%)	ADF (%)	NDF	TDN	RFV	Maint	Lact	Gain
245274	3876	1/10/2001		9.6	10.8							
245154	3874	1/3/2001		10.2	19.2	39.7		57.9		0.56	0.59	0.31
245155	3875	1/3/2001		11.3	22.5	30.4		65.2		0.67	0.67	0.40

Figure 15 : Select Texture test and a specific sample number



Choose the Samples to View Results

Test Type:

- ● Fertility
- ○ Water & Salinity
- ○ Forage
- ○ Texture

Result Set:

- ● Most recent day
- ○ Chose days from: ___ to ___
- ○ Specific sample No.: ___
- ○ All samples

Customer: 803 Sample No.: 33 Texture Testing Result

LabID	Sample	DateReceived	Texture Class	Sand (%)	Clay (%)	Silt (%)
250396	33	3/19/2001	Clay Loam	20.0	30.0	50.0

4.4 Report

From the selected result table, a client may choose any sample by clicking the LabID to view the corresponding report. The data needed to generate a Fertility Report or a Forage Report comes from the Login table, the CustomerInformation table, and the Fertility or Forage table. If a crop code is available for this sample, then the YGandCode (Crop Code and Yield Goal) table also is needed for the report. The Water & Salinity and Texture tests only need three tables to generate a corresponding report: the Login table, the CustomerInformation table, the Salinity or Texture table. A soil sample may request Fertility, or Salinity or Texture analysis. Sometimes it requests all of them; thus, for one sample there is possibly more than one report.

A soil, water or forage test is of no value unless the results are interpreted properly. OSU spent many years conducting research to determine the proper interpretations of the tests performed by the SWFAL [McCray, B.J., 2000]. The interpretation is generated according to the analysis of each of the individual sample results; many data are not visible to the end users. There are four functions to generate reports; each one handles a specific testing type. For Water/Salinity testing, there are five sub-testing types. Therefore, five subroutines produce Water & Salinity reports.

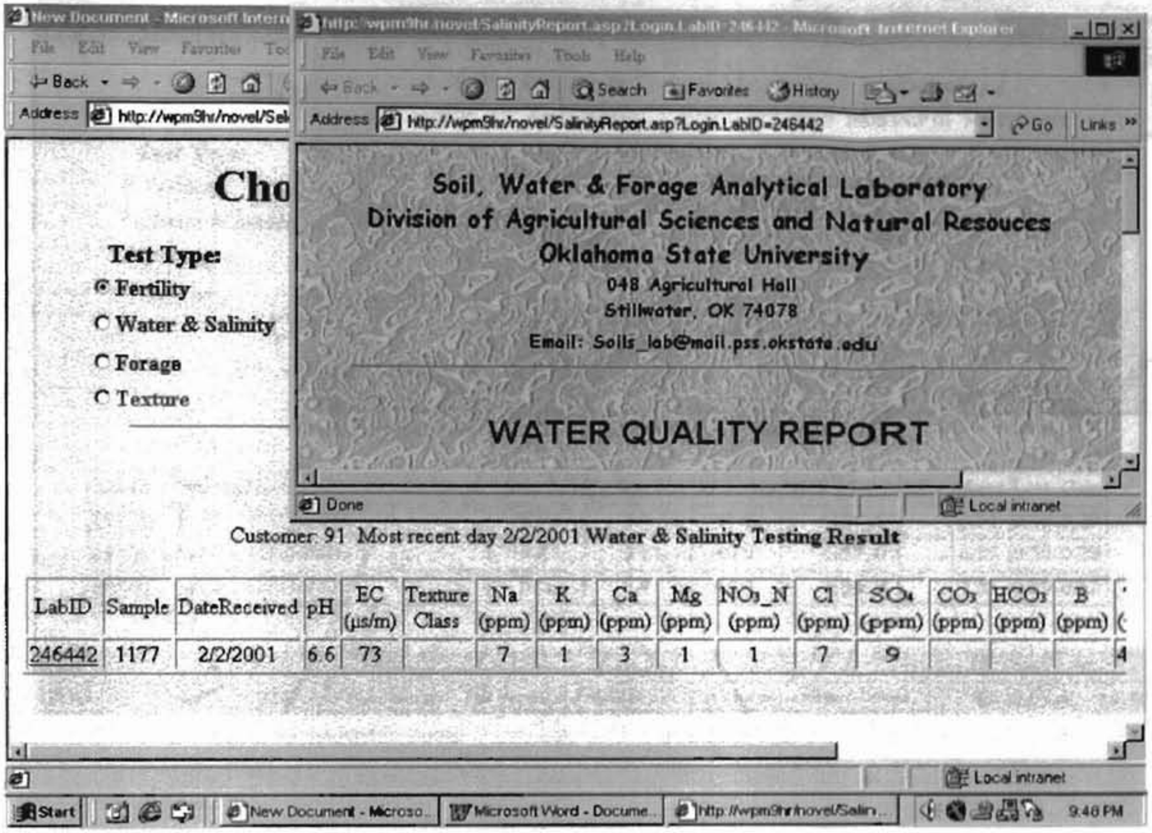
Table 2 shows which samples require interpretation. Figure 16 and Figure 17 present the pop-up report pages.

Table 2: Reports request Interpretations and Recommendations:

Test Type	**Sub Test**	**Interpretation and Recommendation**
Soil Test (for 36 crops)	No crop provide	No
	Crop 32 (tree) Crop 33 (rose)	No
	Other crops	Yes
Water & Salinity Test	Water for irrigation	Yes
	Water for livestock	Yes
	Water for household	Static Recommendation
	Salinity management	Yes
	Comprehensive salinity	Yes
Forage Test	Crop 22 (Sorghum-Sudan Hay)	Yes
	Other crop	No
Texture Test		No

re report page

Figure 16: Water & Salinity report page


New Document - Microsoft Intern
File Edit View Favorites Too
Back
Address http://wpm9hr/novel/Sel
http://wpm9hr/novel/SalinityReport.asp?Login.LabID=246442 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://wpm9hr/novel/SalinityReport.asp?Login.LabID=246442
Go Links
Done
Local intranet
Start
New Document - Microso...
Microsoft Word - Docume...
http://wpm9hr/novel/Salin...
9:48 PM

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall
Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

WATER QUALITY REPORT

Cho

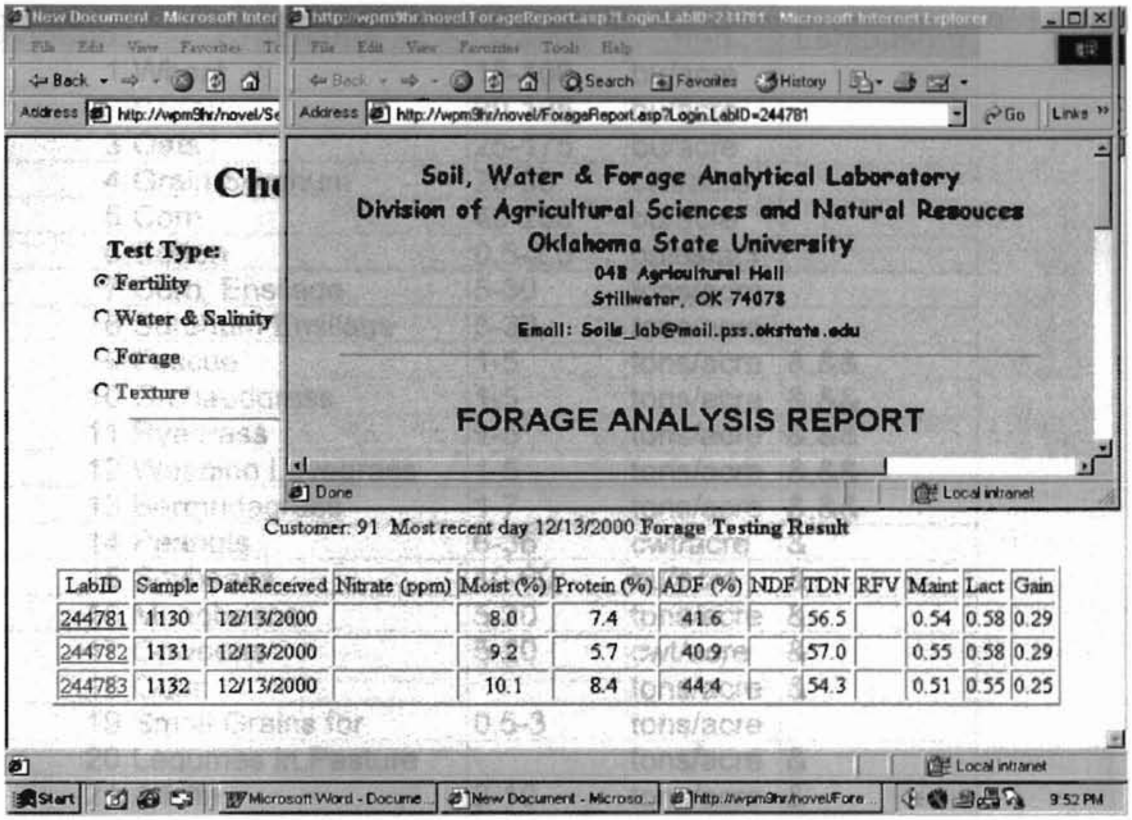
Test Type:

● Fertility
○ Water & Salinity
○ Forage
○ Texture

Customer: 91 Most recent day 2/2/2001 Water & Salinity Testing Result

LabID	Sample	DateReceived	pH	EC (µs/m)	Texture Class	Na (ppm)	K (ppm)	Ca (ppm)	Mg (ppm)	NO_3_N (ppm)	Cl (ppm)	SO_4 (ppm)	CO_3 (ppm)	HCO_3 (ppm)	B (ppm)	' (
246442	1177	2/2/2001	6.6	73		7	1	3	1	1	7	9				4

Figure 17: Forage report page


New Document - Microsoft Inter
http://wpm9hr/novel/ForageReport.asp?Login.LabID=244781 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
History
Address http://wpm9hr/novel/S
Address http://wpm9hr/novel/ForageReport.asp?Login.LabID=244781
Go
Links
Ch
Done
Local intranet
Start
Microsoft Word - Docume...
New Document - Microso...
http://wpm9hr/novel/Fore...
9:52 PM

Test Type:

● Fertility
○ Water & Salinity
○ Forage
○ Texture

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall
Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

FORAGE ANALYSIS REPORT

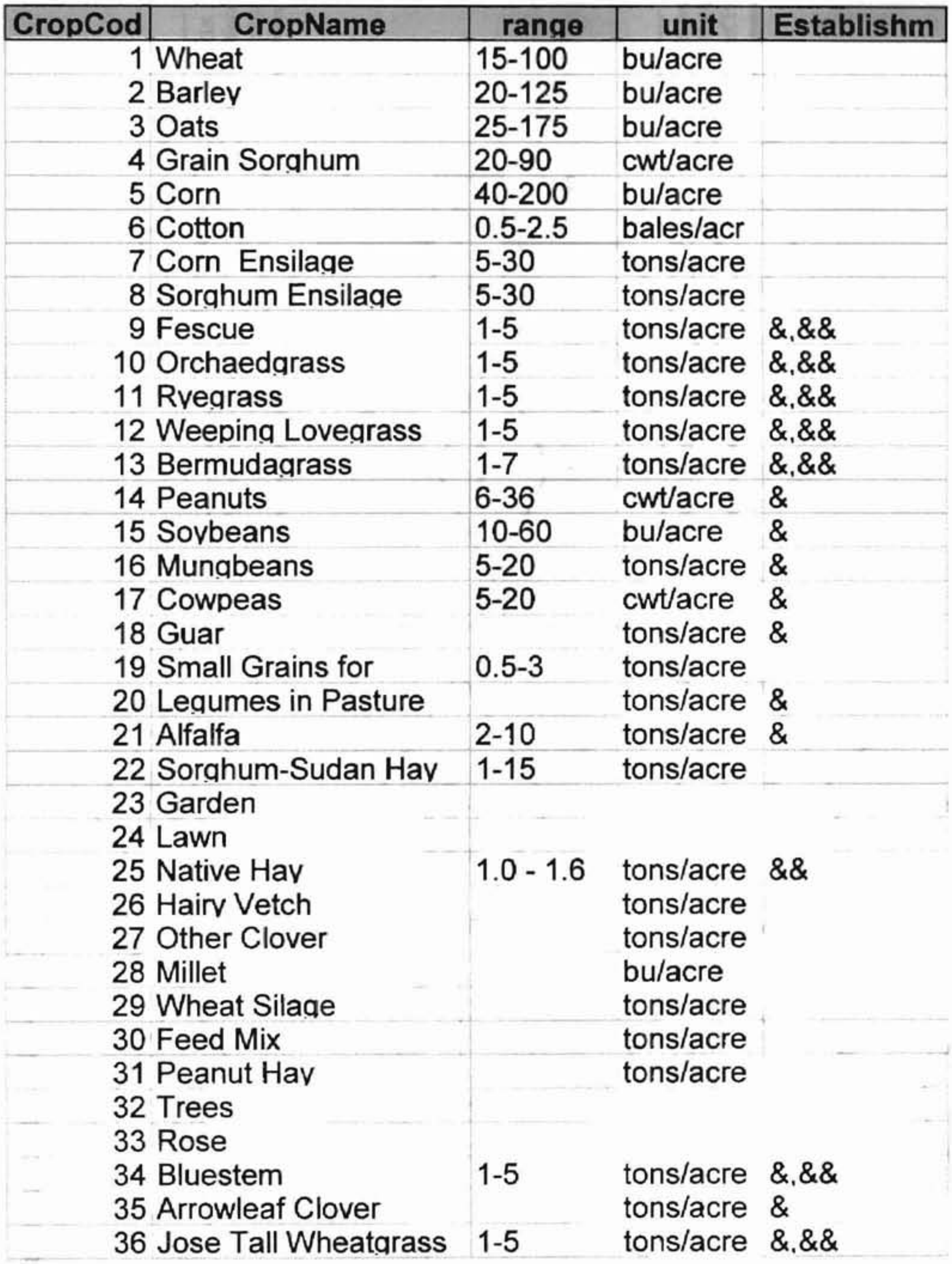
Customer: 91 Most recent day 12/13/2000 **Forage Testing Result**

LabID	Sample	DateReceived	Nitrate (ppm)	Moist (%)	Protein (%)	ADF (%)	NDF	TDN	RFV	Maint	Lact	Gain
244781	1130	12/13/2000		8.0	7.4	41.6		56.5		0.54	0.58	0.29
244782	1131	12/13/2000		9.2	5.7	40.9		57.0		0.55	0.58	0.29
244783	1132	12/13/2000		10.1	8.4	44.4		54.3		0.51	0.55	0.25

- **Fertility Report**

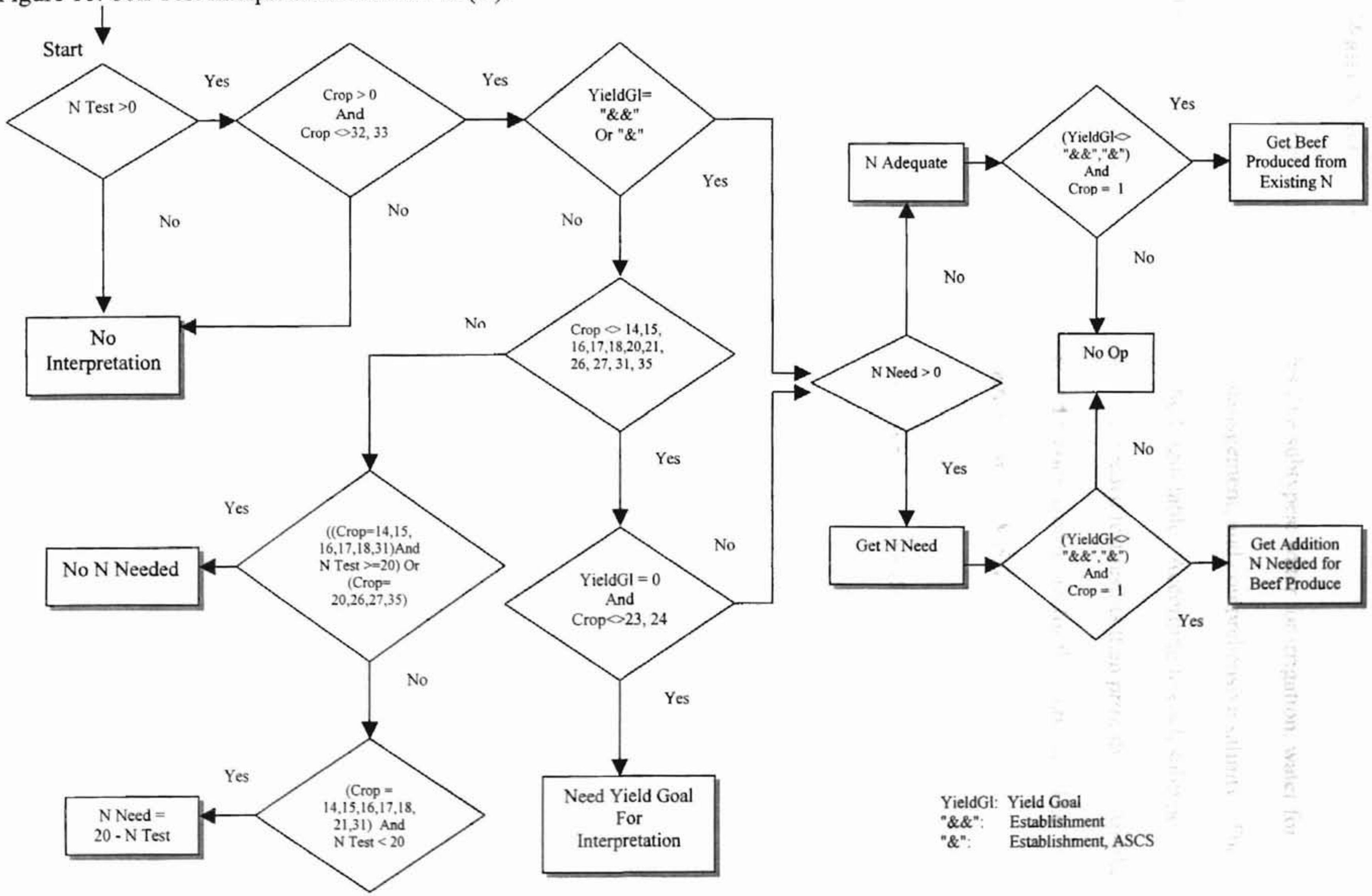
Most soil sample interpretations depend upon what plants the customer intends to grow. Currently SWFAL provides information for 36 crops to which the customers may relate for their soil samples. Table 3 is the list of these crops. For each soil sample, a customer may select any test item from following list: pH, Buffer Index, NO_3-N, K, P, SO_4, Ca, Na, Mg, Cl, Fe, Zn, B (a list of chemical elements, compounds can be found in appendix). When the results are entered into the database and verified to be correct, then they can be processed by the ASP program to generate interpretations and recommendations. Figure 18 uses one test item -- NO_3-N to illustrate the interpretation processing. Figure 21 presents a Web report for Fertility test.

Table 3: YGandCode (Crop code and yield goal) Table



CropCod	CropName	range	unit	Establishm
1	Wheat	15-100	bu/acre	
2	Barley	20-125	bu/acre	
3	Oats	25-175	bu/acre	
4	Grain Sorghum	20-90	cwt/acre	
5	Corn	40-200	bu/acre	
6	Cotton	0.5-2.5	bales/acr	
7	Corn Ensilage	5-30	tons/acre	
8	Sorghum Ensilage	5-30	tons/acre	
9	Fescue	1-5	tons/acre	&,&&
10	Orchaedgrass	1-5	tons/acre	&,&&
11	Ryegrass	1-5	tons/acre	&,&&
12	Weeping Lovegrass	1-5	tons/acre	&,&&
13	Bermudagrass	1-7	tons/acre	&,&&
14	Peanuts	6-36	cwt/acre	&
15	Soybeans	10-60	bu/acre	&
16	Mungbeans	5-20	tons/acre	&
17	Cowpeas	5-20	cwt/acre	&
18	Guar		tons/acre	&
19	Small Grains for	0.5-3	tons/acre	
20	Legumes in Pasture		tons/acre	&
21	Alfalfa	2-10	tons/acre	&
22	Sorghum-Sudan Hay	1-15	tons/acre	
23	Garden			
24	Lawn			
25	Native Hay	1.0 - 1.6	tons/acre	&&
26	Hairy Vetch		tons/acre	
27	Other Clover		tons/acre	
28	Millet		bu/acre	
29	Wheat Silage		tons/acre	
30	Feed Mix		tons/acre	
31	Peanut Hay		tons/acre	
32	Trees			
33	Rose			
34	Bluestem	1-5	tons/acre	&,&&
35	Arrowleaf Clover		tons/acre	&
36	Jose Tall Wheatgrass	1-5	tons/acre	&,&&

Figure 18: Soil Test Interpretation for NO3-N (N):



- **Water & Salinity Report**

A Water & Salinity test includes five subtypes: water for irrigation, water for livestock, water for household, salinity management, and comprehensive salinity. The subtype information can be found from the Login table. According to each subtype, a corresponding function is called to query the needed test results then print them. Another function will handle the interpretations and recommendations for this subtype. The following functions are included to generate a Water & Salinity Report:

```
<!-- METADATA TYPE = "typelib"
    FILE = "C:\Program Files\Common Files\System\ado\msado15.dll" -->
<!-- #include file= "ReportHead.asp" -->
<!-- #include file= "DataStore.asp" -->
<!-- #include file= "ReportTitle.asp" -->
<!-- #include file= "Water1.asp" -->
<!-- #include file= "Water2.asp" -->
<!-- #include file= "Water3.asp" -->
<!-- #include file= "Salinity5.asp" -->
<!-- #include file= "Salinity6.asp" -->
<!-- #include file= "Inter1.asp" -->
<!-- #include file= "Inter2.asp" -->
<!-- #include file= "Inter3.asp" -->
<!-- #include file= "Inter5.asp" -->
<!-- #include file= "Inter6.asp" -->
```

The function of interpretation for one of the subtest Salinity Management, is implemented with the following VBScript code (the interpretation text definition part from TextA through TextG is omitted):

```
<html>

<%

Function Inter5(objRS)                        'function takes a record set object as input parameter

dim strS, ECC, Texture, ESP, NoInter          ' variables to receive objRS field values

dim EspV, GYPSUM, saltAmt, Wilt               ' for calculated values

dim TextA, TextB, TextC, TextD, TextE, TextE1, TextF, TextG  ' interpretation text

dim strSQL, objRSgyp, objConn                 ' objects and query string

ECC = objRS.Fields("ECC")                     ' get value from parameter object

Texture = objRS.Fields("Texture")

ESP = objRS.Fields("ESP")

NoInter = objRS.Fields("NoInter")

GYPSUM = objRS.Fields("GYPSUM")

if Texture = "F" or Texture = "f" then        'interpretation for texture calss

    Texture = "Fine"

elseif Texture = "C" or Texture = "c" then

    Testure = "Coarse"

elseif Texture = "M" or Texture = "m" then

    Texture = "Medium"

end if

if ECC <= 4000 then                           ' identify crop catagory

     Wilt = "sensitive crops."

elseif 4000 < ECC and ECC <= 12000 then

    Wilt = "moderately tolerant crops."

elseif 12000 < ECC then

 Wilt = "even salt tolerant crops."

end if

saltAmt = CInt(ECC/4000)                      ' salt amount

'***********
```

```
' here omit all the codes assigning the values of  text from "TextA" to "TextG"
'***********
strS = strS & "<UL><font size = '2'>"                        'start an unnumbered list
if 8000 < ECC then
    strS = strS & "<LI>" & TextC & "</LI>"
 if ESP <= 15 then
     strS = strS & "<LI>" & TextF & "&nbsp;" & TextG & "</LI>"
 elseif 15 < ESP then
    strS = strS & "<LI>" & TextD & "&nbsp;" & TextF & "</LI>"
    if GYPSUM <= 5 then
      strS = strS & "<LI>" & TextE & "</LI>"
    elseif GYPSUM > 5 then
      strS = strS & "<LI>" & TextE1 & "</LI>"
    end if
    strS = strS & "<LI>" & TextG  & "</LI>"
end if
elseif 4000 <= ECC and ECC <= 8000 then
    strS = strS & "<LI>" & TextB & "</LI>"
    if ESP <= 15 then
       strS = strS & "<LI>" & TextF & "&nbsp;" & TextG & "</LI>"
    elseif 15 < ESP then
      strS = strS & "<LI>" & TextD & "&nbsp;" & TextF & "</LI>"
      if GYPSUM <= 5 then
        strS = strS & "<LI>" & TextE & "</LI>"
     else
       strS = strS & "<LI>" & TextE1 & "</LI>"
     end if
     strS = strS & "<LI>" & TextG & "</LI>"
    end if
elseif ECC < 4000 then
    strS = strS & "<LI>" & TextA  & "</LI>"
    if 15 < ESP then
      strS = strS & "<LI>" & TextD & "&nbsp;" & TextF & "</LI>"
      if GYPSUM <= 5 then
         strS = strS & "<LI>" & TextE & "</LI>"
      else
         strS = strS & "<LI>" & TextE1 & "</LI>"
      end if
      strS = strS & "<LI>" & TextG & "</LI>"
end if
end if
strS = strS & "</font></UL>"                    'finish the list
Inter5 = strS                                   'return the function value as a string
End Function
%>
</html>
```

For a Water & Salinity Report, some county extension offices prefer to provide their own interpretation to their customers. In this case, SWFAL provides a blank input text box on the report allowing the extension agents to type their own comments. Figure 27 is a sample of this kind of reports. Figure 22 through Figure 26 show the sample report for each Water & Salinity subtest.

- **Forage Report**

A Forage report takes a LabID, queries the required information from the Login table, the CustomerInformation, and the Forage tables (if crop code is available, YgandCode table is also needed to get crop name), then calls ReportHead and ReportTitle to generate a Forage Report. There is only one crop, 22 (sorghum-Sudan hay), that needs an interpretation for this test, the interpretation is coded within the report function, and does not use a separate interpretation function.

- **TextureReport**

This report takes a LabID, queries the sample information from the Login, the Customer and the Texture tables, then calls ReportHead and ReportTitle to generate a Texture Report. A Texture Report does not request interpretation. All its calculations are processed by the Visual Basic program within the DBMS and report needed information are stored in the database. The Web application only needs to abstract the results data from the database, making the Web report for Texture relatively simple. A sample of Texture Report is shown on Figure 29.

4.5 Other important functions:

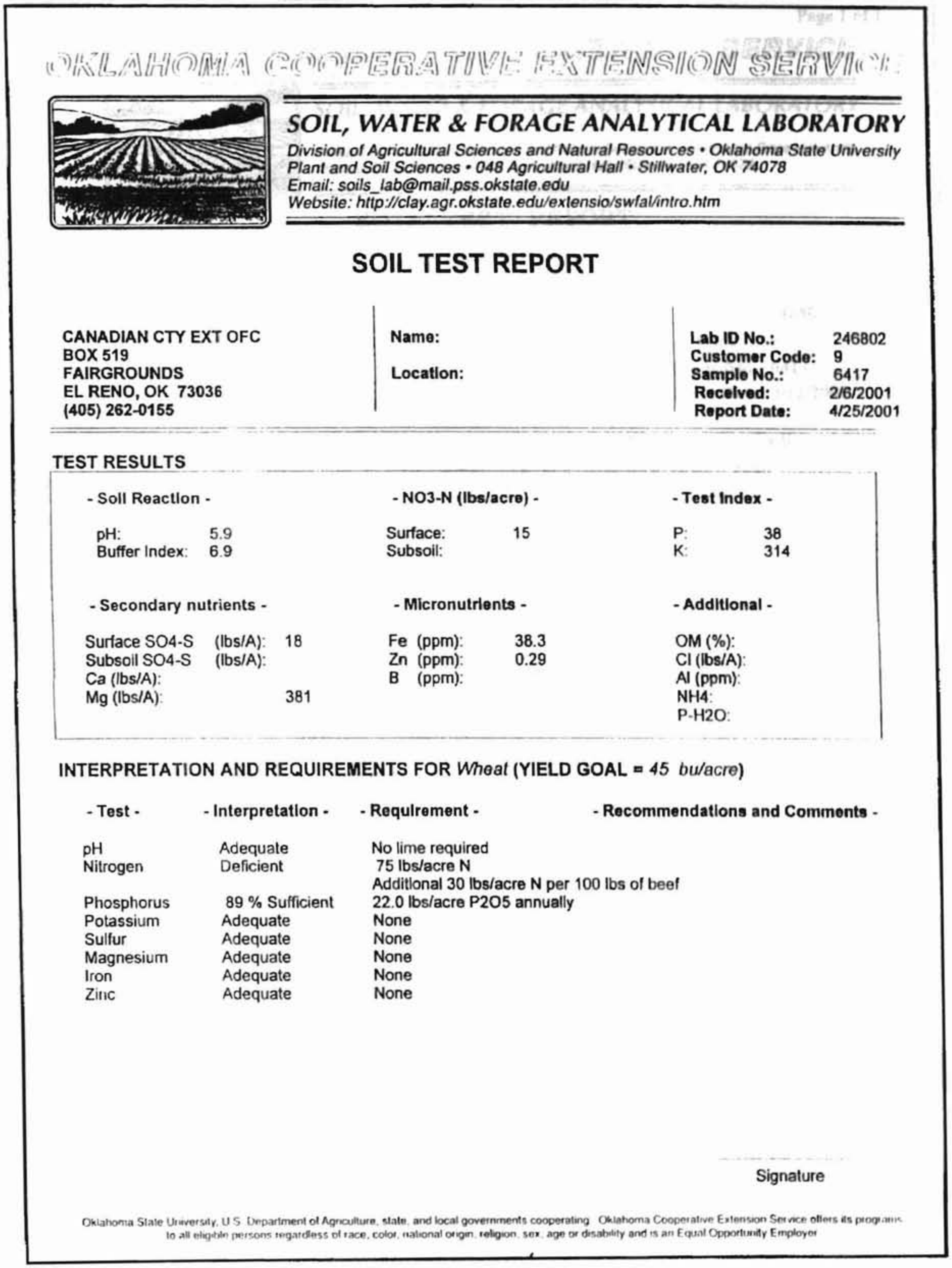
Global.asa

This function keeps application level variables for all users and session level variables for each user. The variable to count the number of visitors is stored in this file. It is modified every time a user accesses the site. Since it is possible for more than one user to access the site at the same time, lock and unlock are used to protect this "critical section" to ensure the count is performed correctly.

DataStore.asp

This function keeps the ADO database connection string. Any page that needs data from the database can include this file to create and open the record set using the same connection. It makes the data retrieving very convenient and fast.

Figure 19: A Hardcopy Report



Page 1 of 1

OKLAHOMA COOPERATIVE EXTENSION SERVICE

SOIL, WATER & FORAGE ANALYTICAL LABORATORY

Division of Agricultural Sciences and Natural Resources • Oklahoma State University
Plant and Soil Sciences • 048 Agricultural Hall • Stillwater, OK 74078
Email: soils_lab@mail.pss.okstate.edu
Website: http://clay.agr.okstate.edu/extensio/swfal/intro.htm

SOIL TEST REPORT

CANADIAN CTY EXT OFC
BOX 519
FAIRGROUNDS
EL RENO, OK 73036
(405) 262-0155

Name:

Location:

Lab ID No.:	246802
Customer Code:	9
Sample No.:	6417
Received:	2/6/2001
Report Date:	4/25/2001

TEST RESULTS

- Soil Reaction -		**- NO3-N (lbs/acre) -**		**- Test Index -**	
pH:	5.9	Surface:	15	P:	38
Buffer Index:	6.9	Subsoil:		K:	314
- Secondary nutrients -		**- Micronutrients -**		**- Additional -**	
Surface SO4-S (lbs/A):	18	Fe (ppm):	38.3	OM (%):	
Subsoil SO4-S (lbs/A):		Zn (ppm):	0.29	Cl (lbs/A):	
Ca (lbs/A):		B (ppm):		Al (ppm):	
Mg (lbs/A):	381			NH4:	
				P-H2O:	

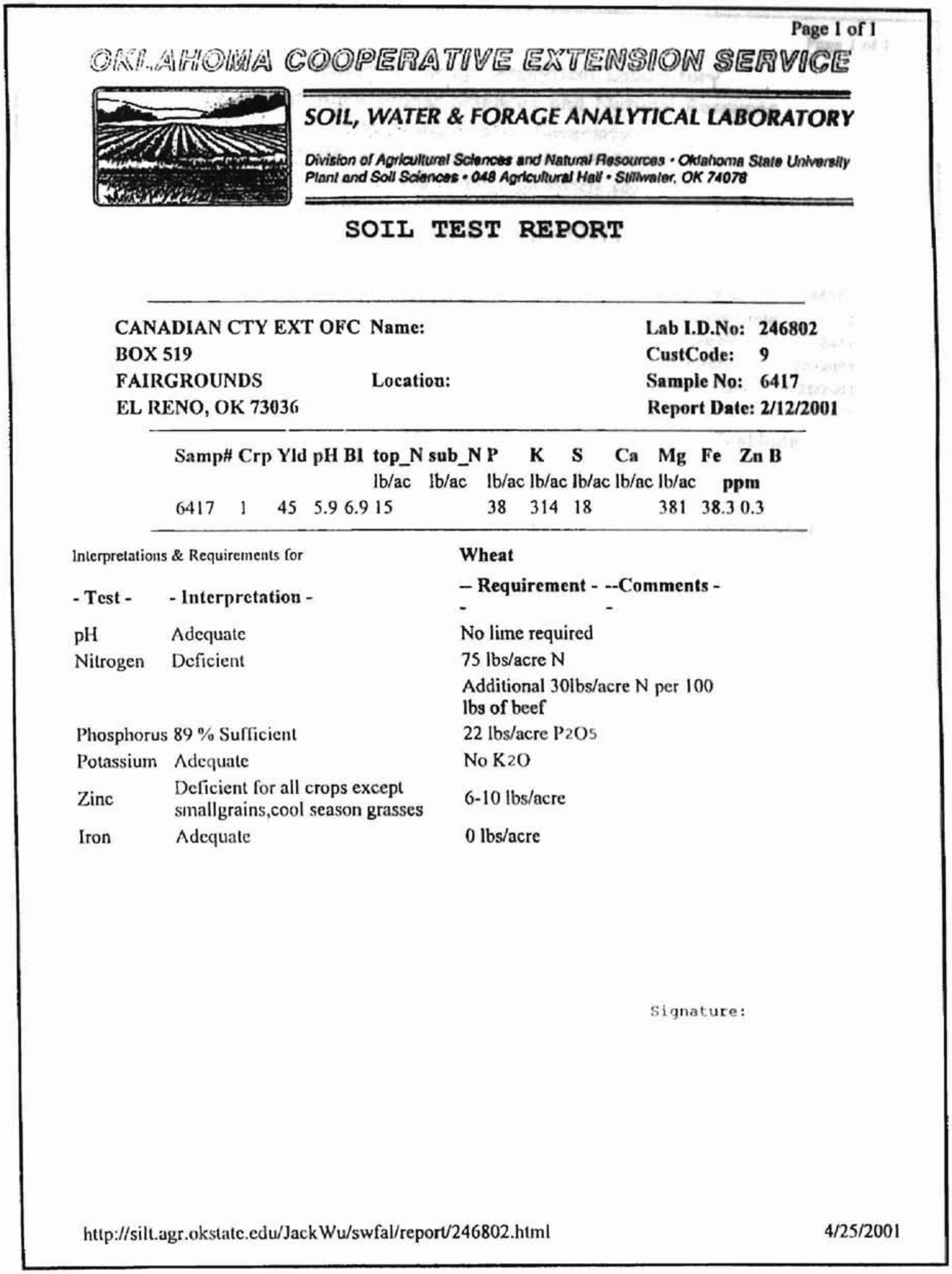
INTERPRETATION AND REQUIREMENTS FOR *Wheat* (YIELD GOAL = *45 bu/acre*)

- Test -	- Interpretation -	- Requirement -	- Recommendations and Comments -
pH	Adequate	No lime required	
Nitrogen	Deficient	75 lbs/acre N Additional 30 lbs/acre N per 100 lbs of beef	
Phosphorus	89 % Sufficient	22.0 lbs/acre P2O5 annually	
Potassium	Adequate	None	
Sulfur	Adequate	None	
Magnesium	Adequate	None	
Iron	Adequate	None	
Zinc	Adequate	None	

Signature

Oklahoma State University, U S Department of Agriculture, state, and local governments cooperating Oklahoma Cooperative Extension Service offers its programs to all eligible persons regardless of race, color, national origin, religion, sex, age or disability and is an Equal Opportunity Employer

Figure 20: A previous Web report



Page 1 of 1

OKLAHOMA COOPERATIVE EXTENSION SERVICE

SOIL, WATER & FORAGE ANALYTICAL LABORATORY

Division of Agricultural Sciences and Natural Resources • Oklahoma State University
Plant and Soil Sciences • 048 Agricultural Hall • Stillwater, OK 74078

SOIL TEST REPORT

CANADIAN CTY EXT OFC Name:
BOX 519
FAIRGROUNDS Location:
EL RENO, OK 73036

Lab I.D.No: 246802
CustCode: 9
Sample No: 6417
Report Date: 2/12/2001

Samp#	Crp	Yld	pH	BI	top_N	sub_N	P	K	S	Ca	Mg	Fe	Zn	B
					lb/ac	lb/ac	lb/ac	lb/ac	lb/ac	lb/ac	lb/ac	ppm		
6417	1	45	5.9	6.9	15		38	314	18		381	38.3	0.3	

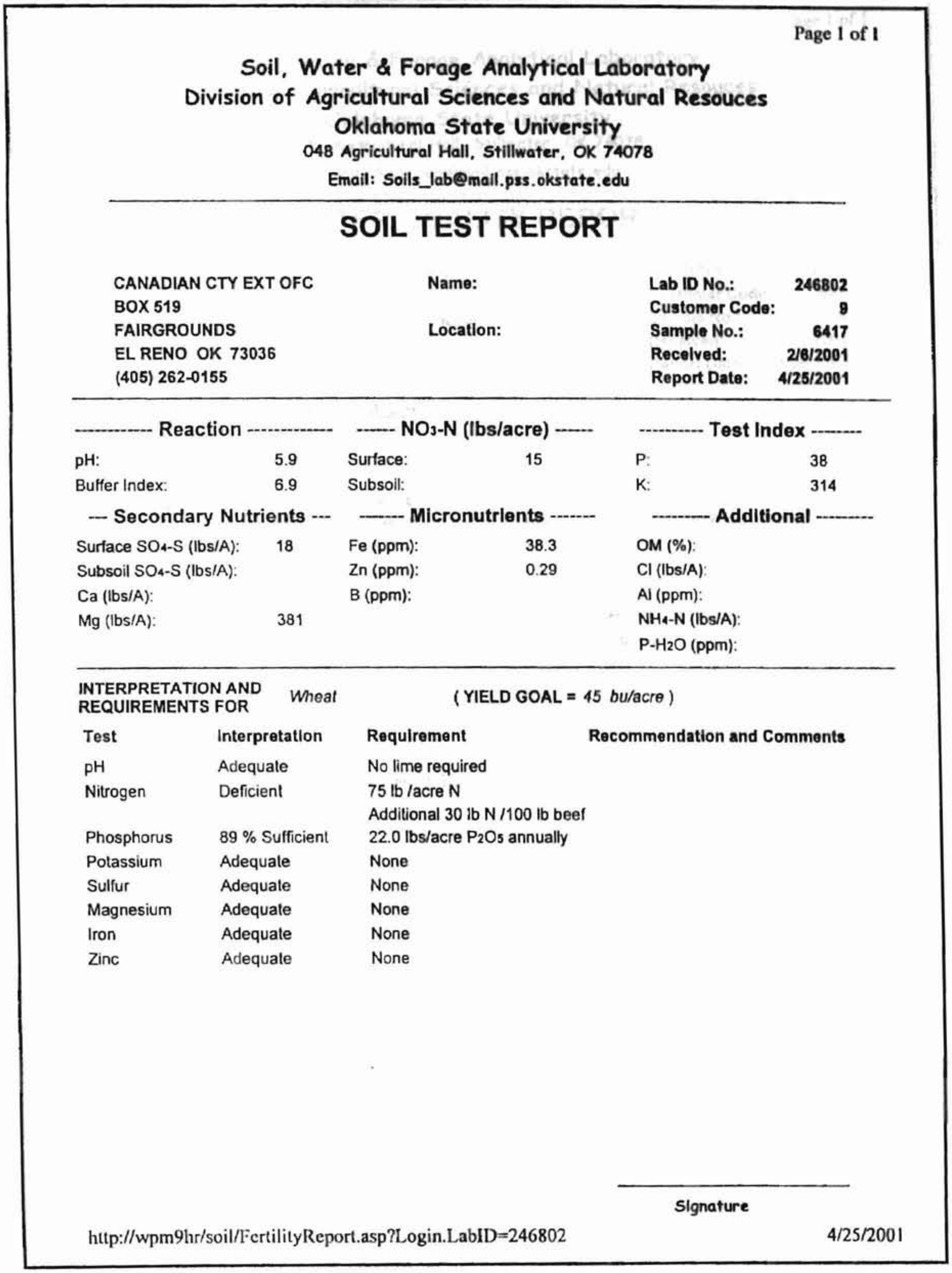
Interpretations & Requirements for **Wheat**

- Test -	- Interpretation -	– Requirement - -	--Comments - -
pH	Adequate	No lime required	
Nitrogen	Deficient	75 lbs/acre N Additional 30lbs/acre N per 100 lbs of beef	
Phosphorus	89 % Sufficient	22 lbs/acre P2O5	
Potassium	Adequate	No K2O	
Zinc	Deficient for all crops except smallgrains,cool season grasses	6-10 lbs/acre	
Iron	Adequate	0 lbs/acre	

Signature:

http://silt.agr.okstate.edu/JackWu/swfal/report/246802.html 4/25/2001

Figure 21: A Web report for Fertility test



Page 1 of 1

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall, Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

SOIL TEST REPORT

CANADIAN CTY EXT OFC	Name:	Lab ID No.:	246802
BOX 519		Customer Code:	9
FAIRGROUNDS	Location:	Sample No.:	6417
EL RENO OK 73036		Received:	2/6/2001
(405) 262-0155		Report Date:	4/25/2001

----------- Reaction ------------		------ NO_3-N (lbs/acre) ------		--------- Test Index --------	
pH:	5.9	Surface:	15	P:	38
Buffer Index:	6.9	Subsoil:		K:	314
--- Secondary Nutrients ---		------- Micronutrients -------		--------- Additional ---------	
Surface SO_4-S (lbs/A):	18	Fe (ppm):	38.3	OM (%):	
Subsoil SO_4-S (lbs/A):		Zn (ppm):	0.29	Cl (lbs/A):	
Ca (lbs/A):		B (ppm):		Al (ppm):	
Mg (lbs/A):	381			NH_4-N (lbs/A):	
				P-H_2O (ppm):	

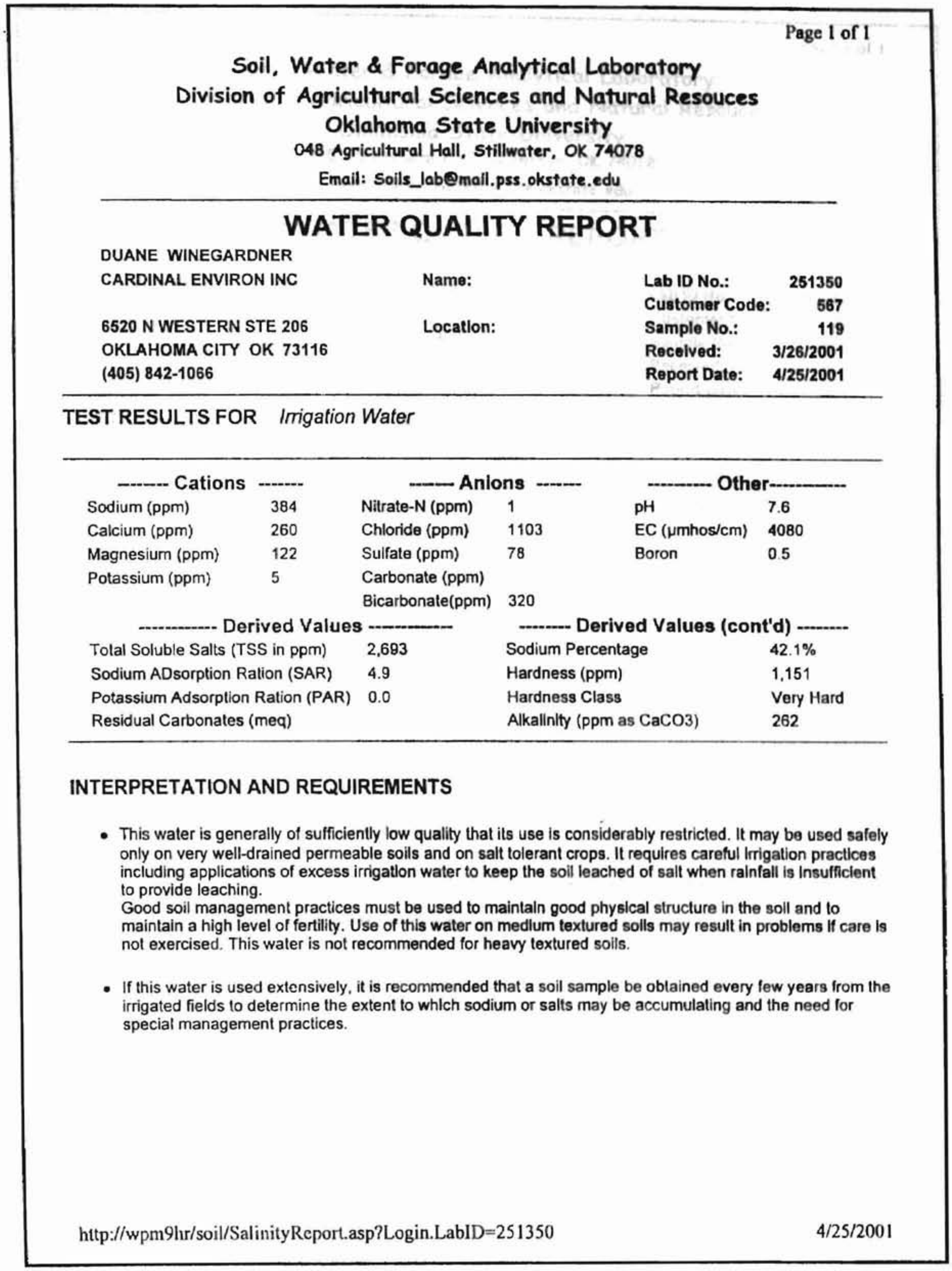
INTERPRETATION AND REQUIREMENTS FOR *Wheat* (YIELD GOAL = *45 bu/acre*)

Test	Interpretation	Requirement	Recommendation and Comments
pH	Adequate	No lime required	
Nitrogen	Deficient	75 lb /acre N Additional 30 lb N /100 lb beef	
Phosphorus	89 % Sufficient	22.0 lbs/acre P_2O_5 annually	
Potassium	Adequate	None	
Sulfur	Adequate	None	
Magnesium	Adequate	None	
Iron	Adequate	None	
Zinc	Adequate	None	

Signature

http://wpm9hr/soil/FertilityReport.asp?Login.LabID=246802 4/25/2001

Figure 22: A Web report for Irrigation Water test



Page 1 of 1

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall, Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

WATER QUALITY REPORT

DUANE WINEGARDNER			
CARDINAL ENVIRON INC	Name:	Lab ID No.:	251350
		Customer Code:	567
6520 N WESTERN STE 206	Location:	Sample No.:	119
OKLAHOMA CITY OK 73116		Received:	3/26/2001
(405) 842-1066		Report Date:	4/25/2001

TEST RESULTS FOR *Irrigation Water*

------- Cations -------		------ Anions -------		---------- Other------------	
Sodium (ppm)	384	Nitrate-N (ppm)	1	pH	7.6
Calcium (ppm)	260	Chloride (ppm)	1103	EC (µmhos/cm)	4080
Magnesium (ppm)	122	Sulfate (ppm)	78	Boron	0.5
Potassium (ppm)	5	Carbonate (ppm)			
		Bicarbonate(ppm)	320		

------------ Derived Values ------------		-------- Derived Values (cont'd) --------	
Total Soluble Salts (TSS in ppm)	2,693	Sodium Percentage	42.1%
Sodium ADsorption Ration (SAR)	4.9	Hardness (ppm)	1,151
Potassium Adsorption Ration (PAR)	0.0	Hardness Class	Very Hard
Residual Carbonates (meq)		Alkalinity (ppm as CaCO3)	262

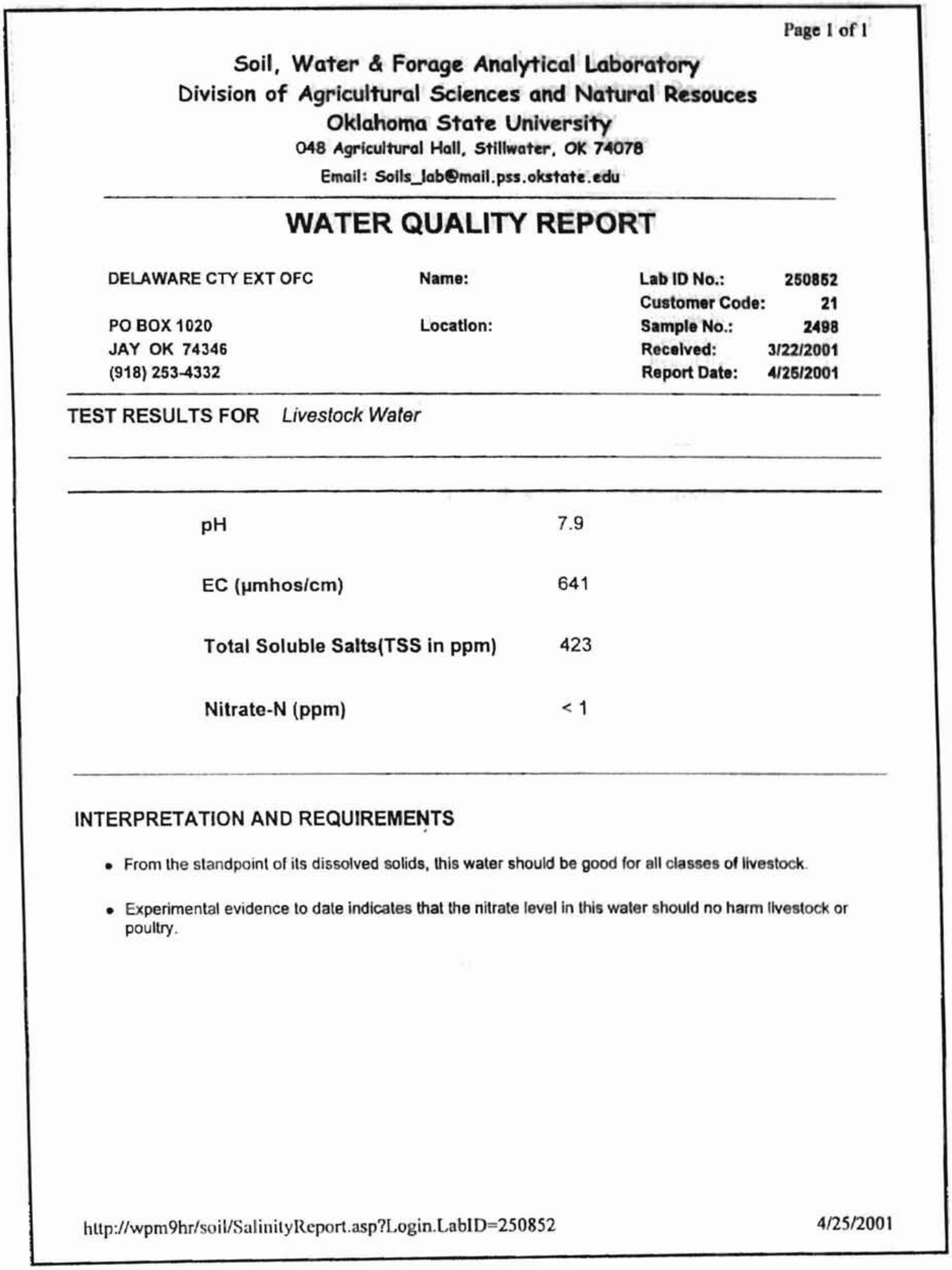
INTERPRETATION AND REQUIREMENTS

- This water is generally of sufficiently low quality that its use is considerably restricted. It may be used safely only on very well-drained permeable soils and on salt tolerant crops. It requires careful Irrigation practices including applications of excess irrigation water to keep the soil leached of salt when rainfall is insufficient to provide leaching.
  Good soil management practices must be used to maintain good physical structure in the soil and to maintain a high level of fertility. Use of this water on medium textured soils may result in problems if care is not exercised. This water is not recommended for heavy textured soils.

- If this water is used extensively, it is recommended that a soil sample be obtained every few years from the irrigated fields to determine the extent to which sodium or salts may be accumulating and the need for special management practices.

http://wpm9hr/soil/SalinityReport.asp?Login.LabID=251350 4/25/2001

Figure 23: A Web report for Livestock Water test



Page 1 of 1

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall, Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

WATER QUALITY REPORT

DELAWARE CTY EXT OFC	Name:	Lab ID No.:	250852
		Customer Code:	21
PO BOX 1020	Location:	Sample No.:	2498
JAY OK 74346		Received:	3/22/2001
(918) 253-4332		Report Date:	4/25/2001

TEST RESULTS FOR *Livestock Water*

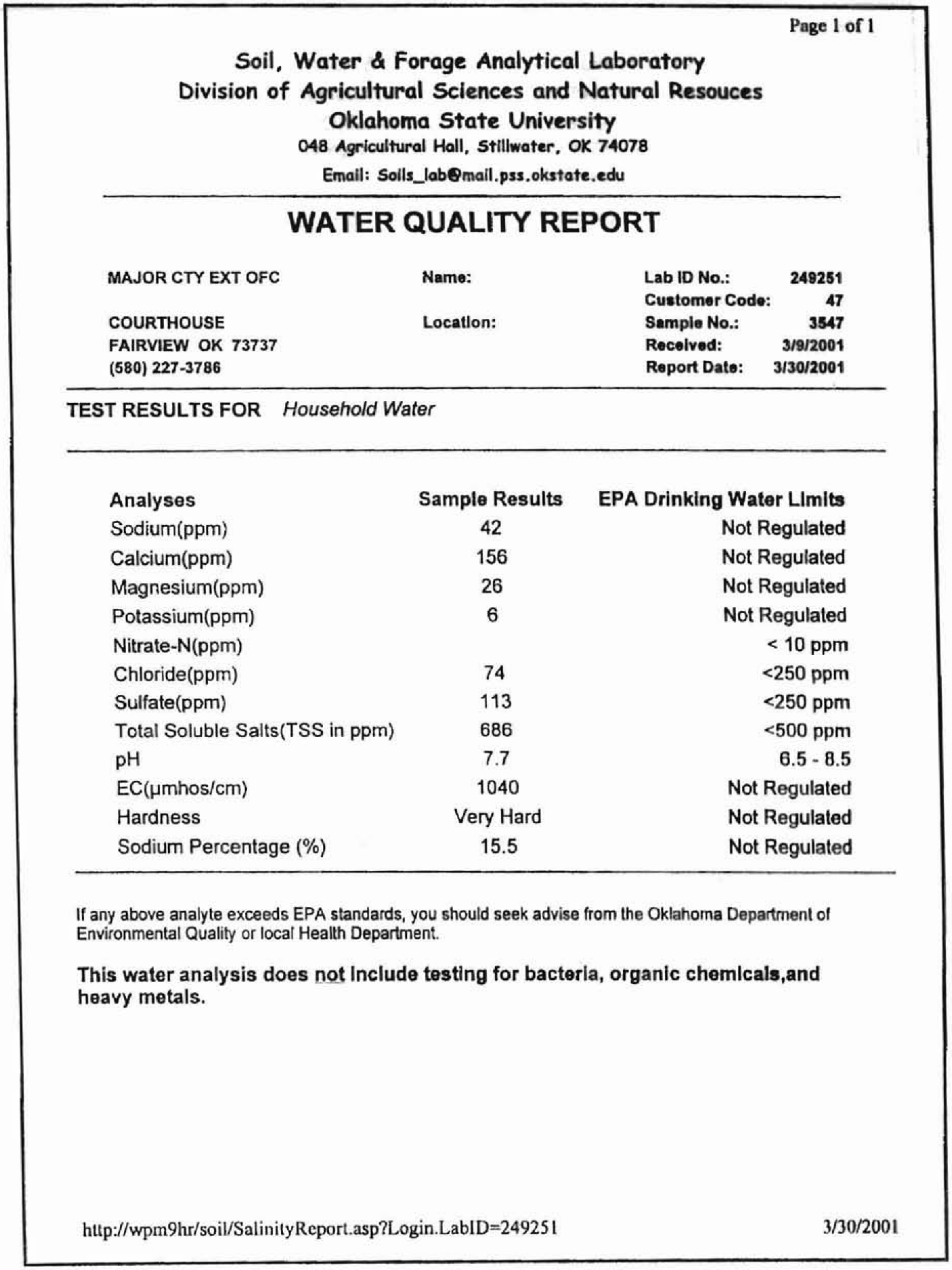
pH	7.9
EC (µmhos/cm)	641
Total Soluble Salts(TSS in ppm)	423
Nitrate-N (ppm)	< 1

INTERPRETATION AND REQUIREMENTS

- From the standpoint of its dissolved solids, this water should be good for all classes of livestock.
- Experimental evidence to date indicates that the nitrate level in this water should no harm livestock or poultry.

http://wpm9hr/soil/SalinityReport.asp?Login.LabID=250852 4/25/2001

Figure 24: A Web report for Household Water test



Page 1 of 1

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall, Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

WATER QUALITY REPORT

MAJOR CTY EXT OFC	Name:	Lab ID No.:	249251
		Customer Code:	47
COURTHOUSE	Location:	Sample No.:	3547
FAIRVIEW OK 73737		Received:	3/9/2001
(580) 227-3786		Report Date:	3/30/2001

TEST RESULTS FOR *Household Water*

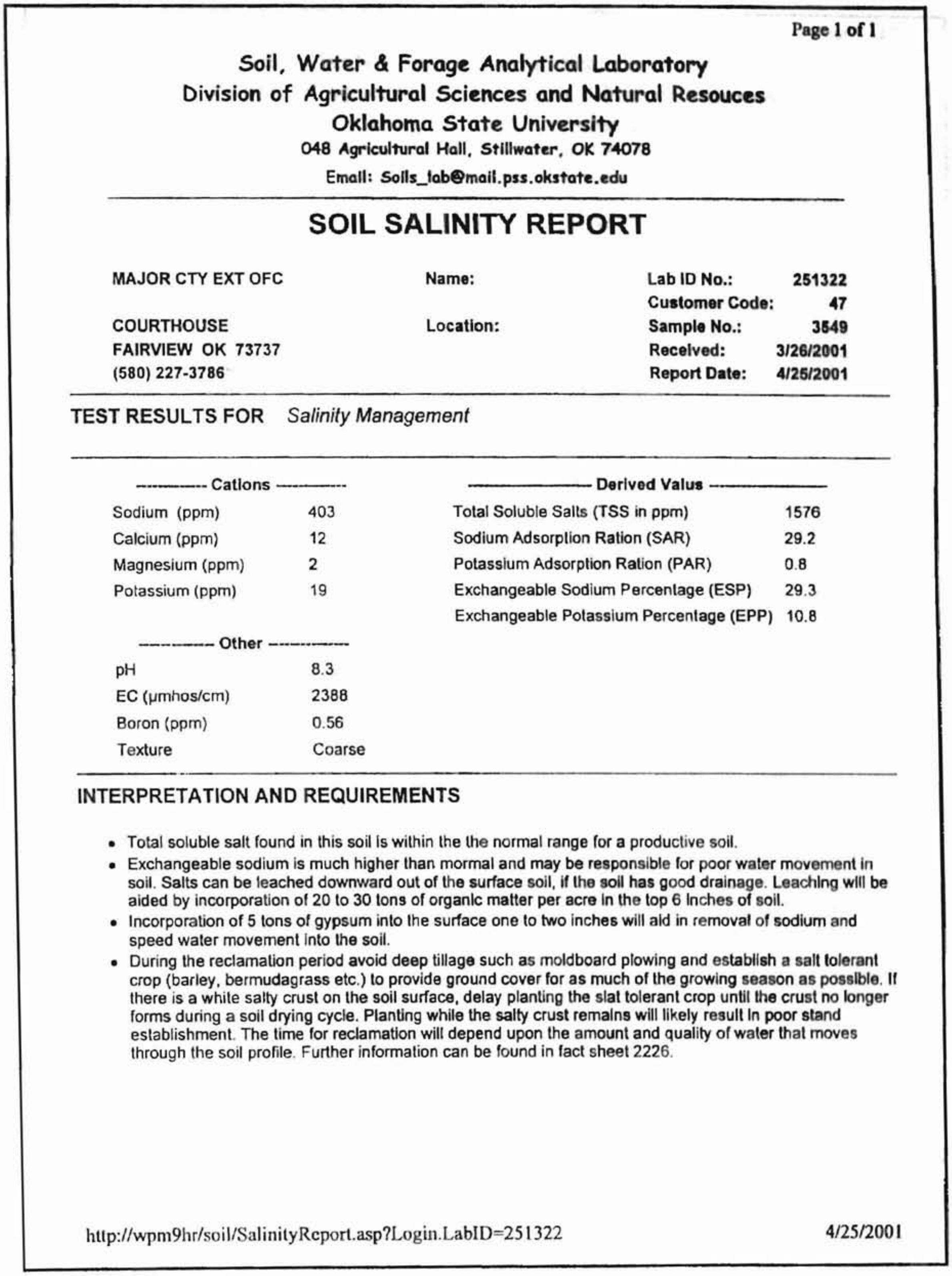
Analyses	Sample Results	EPA Drinking Water Limits
Sodium(ppm)	42	Not Regulated
Calcium(ppm)	156	Not Regulated
Magnesium(ppm)	26	Not Regulated
Potassium(ppm)	6	Not Regulated
Nitrate-N(ppm)		< 10 ppm
Chloride(ppm)	74	<250 ppm
Sulfate(ppm)	113	<250 ppm
Total Soluble Salts(TSS in ppm)	686	<500 ppm
pH	7.7	6.5 - 8.5
EC(μmhos/cm)	1040	Not Regulated
Hardness	Very Hard	Not Regulated
Sodium Percentage (%)	15.5	Not Regulated

If any above analyte exceeds EPA standards, you should seek advise from the Oklahoma Department of Environmental Quality or local Health Department.

This water analysis does not include testing for bacteria, organic chemicals,and heavy metals.

http://wpm9hr/soil/SalinityReport.asp?Login.LabID=249251 3/30/2001

Figure 25: A Web report for Salinity Management test



Page 1 of 1

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall, Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

SOIL SALINITY REPORT

MAJOR CTY EXT OFC	Name:	Lab ID No.:	251322
		Customer Code:	47
COURTHOUSE	Location:	Sample No.:	3549
FAIRVIEW OK 73737		Received:	3/26/2001
(580) 227-3786		Report Date:	4/25/2001

TEST RESULTS FOR *Salinity Management*

---------- Cations ----------	
Sodium (ppm)	403
Calcium (ppm)	12
Magnesium (ppm)	2
Potassium (ppm)	19

---------- Other ----------	
pH	8.3
EC (μmhos/cm)	2388
Boron (ppm)	0.56
Texture	Coarse

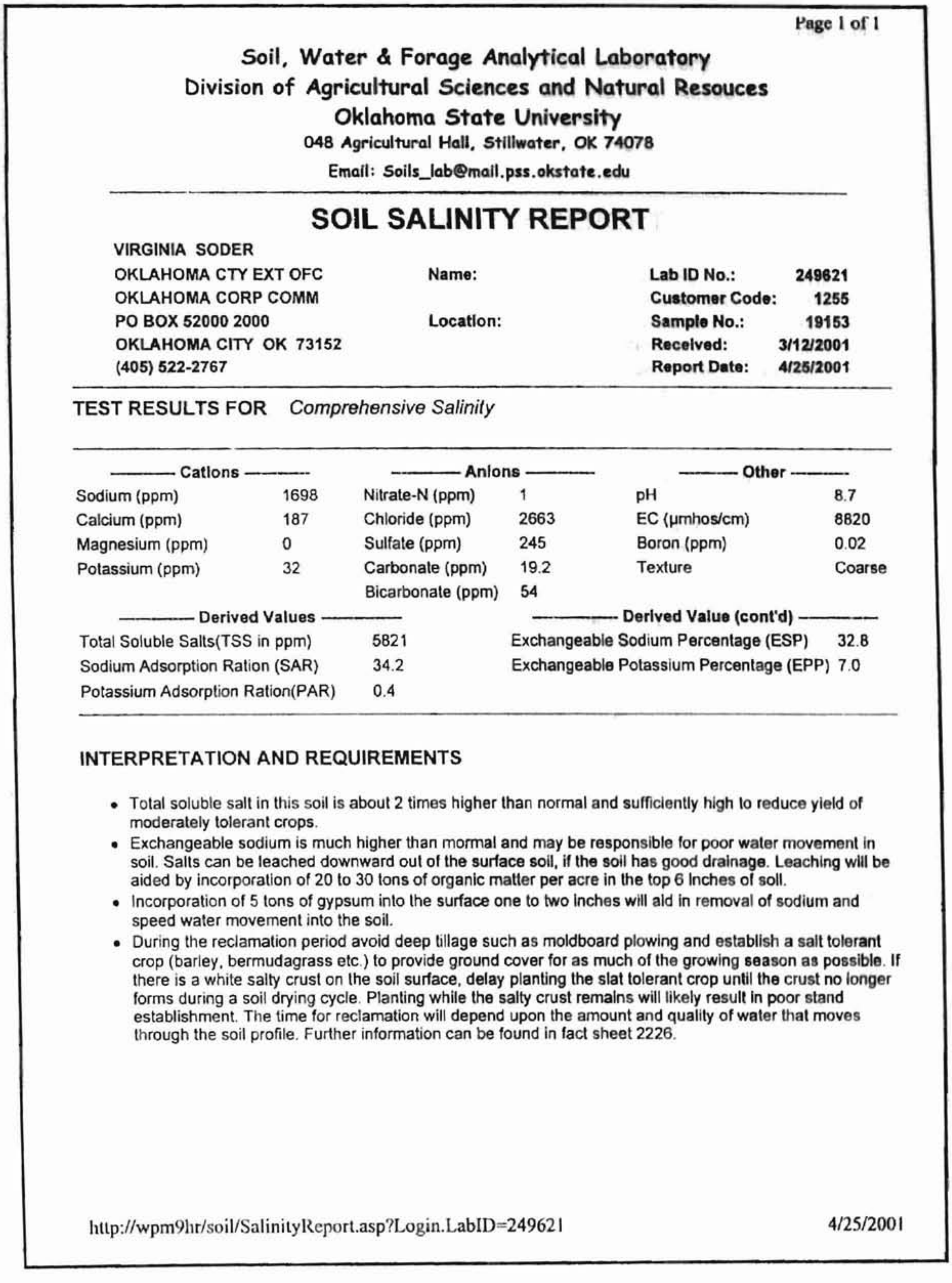
---------- Derived Valus ----------	
Total Soluble Salts (TSS in ppm)	1576
Sodium Adsorption Ration (SAR)	29.2
Potassium Adsorption Ration (PAR)	0.8
Exchangeable Sodium Percentage (ESP)	29.3
Exchangeable Potassium Percentage (EPP)	10.8

INTERPRETATION AND REQUIREMENTS

- Total soluble salt found in this soil is within the the normal range for a productive soil.
- Exchangeable sodium is much higher than mormal and may be responsible for poor water movement in soil. Salts can be leached downward out of the surface soil, if the soil has good drainage. Leaching will be aided by incorporation of 20 to 30 tons of organic matter per acre in the top 6 inches of soil.
- Incorporation of 5 tons of gypsum into the surface one to two inches will aid in removal of sodium and speed water movement into the soil.
- During the reclamation period avoid deep tillage such as moldboard plowing and establish a salt tolerant crop (barley, bermudagrass etc.) to provide ground cover for as much of the growing season as possible. If there is a white salty crust on the soil surface, delay planting the slat tolerant crop until the crust no longer forms during a soil drying cycle. Planting while the salty crust remains will likely result in poor stand establishment. The time for reclamation will depend upon the amount and quality of water that moves through the soil profile. Further information can be found in fact sheet 2226.

http://wpm9hr/soil/SalinityReport.asp?Login.LabID=251322 4/25/2001

Figure 26: A Web report for Comprehensive Salinity test



Page 1 of 1

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall, Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

SOIL SALINITY REPORT

VIRGINIA SODER			
OKLAHOMA CTY EXT OFC	Name:	Lab ID No.:	249621
OKLAHOMA CORP COMM		Customer Code:	1255
PO BOX 52000 2000	Location:	Sample No.:	19153
OKLAHOMA CITY OK 73152		Received:	3/12/2001
(405) 522-2767		Report Date:	4/25/2001

TEST RESULTS FOR *Comprehensive Salinity*

Cations		Anions		Other	
Sodium (ppm)	1698	Nitrate-N (ppm)	1	pH	8.7
Calcium (ppm)	187	Chloride (ppm)	2663	EC (µmhos/cm)	8820
Magnesium (ppm)	0	Sulfate (ppm)	245	Boron (ppm)	0.02
Potassium (ppm)	32	Carbonate (ppm)	19.2	Texture	Coarse
		Bicarbonate (ppm)	54		

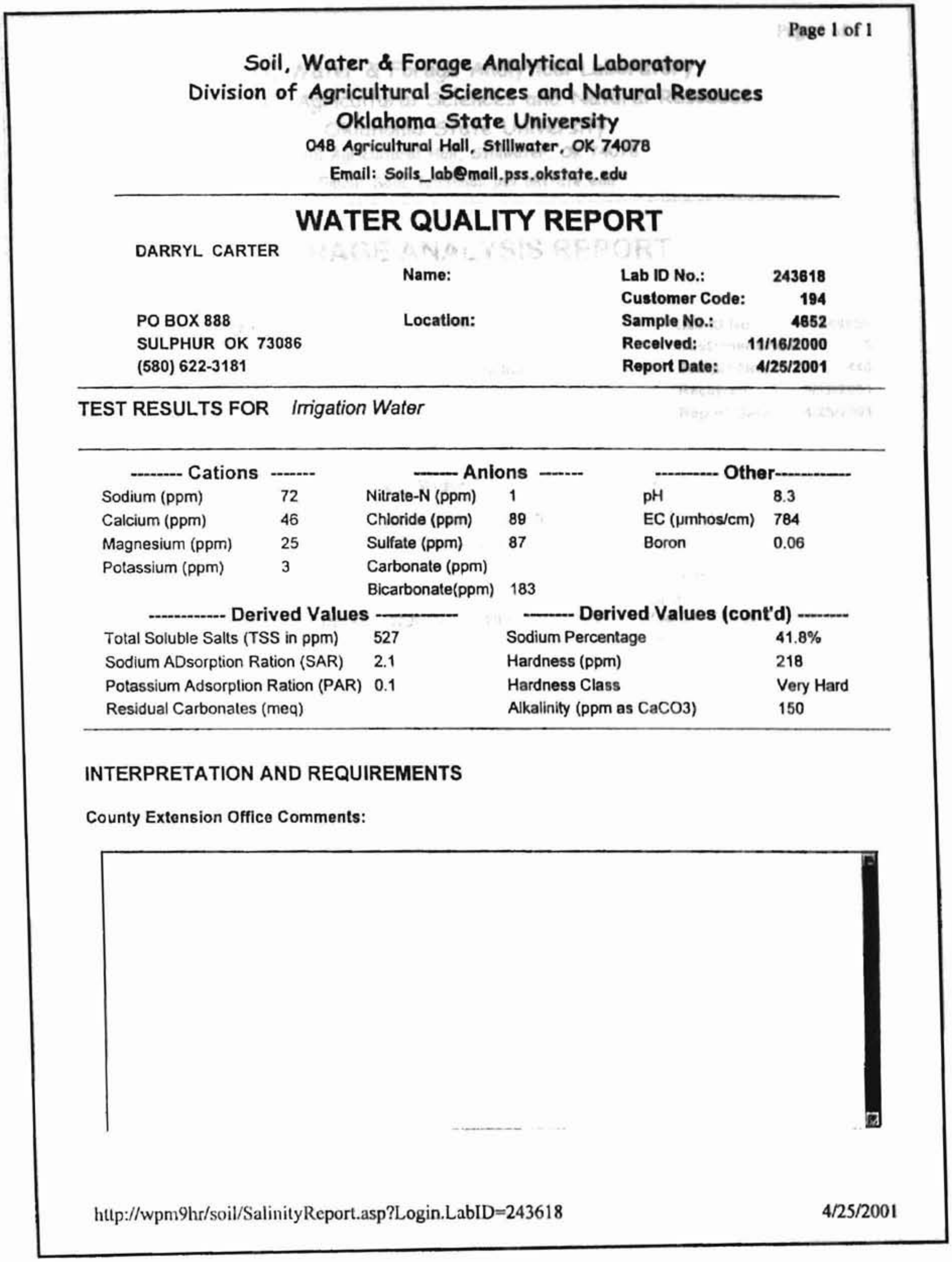
Derived Values		Derived Value (cont'd)	
Total Soluble Salts(TSS in ppm)	5821	Exchangeable Sodium Percentage (ESP)	32.8
Sodium Adsorption Ration (SAR)	34.2	Exchangeable Potassium Percentage (EPP)	7.0
Potassium Adsorption Ration(PAR)	0.4		

INTERPRETATION AND REQUIREMENTS

- Total soluble salt in this soil is about 2 times higher than normal and sufficiently high to reduce yield of moderately tolerant crops.
- Exchangeable sodium is much higher than mormal and may be responsible for poor water movement in soil. Salts can be leached downward out of the surface soil, if the soil has good drainage. Leaching will be aided by incorporation of 20 to 30 tons of organic matter per acre in the top 6 inches of soil.
- Incorporation of 5 tons of gypsum into the surface one to two inches will aid in removal of sodium and speed water movement into the soil.
- During the reclamation period avoid deep tillage such as moldboard plowing and establish a salt tolerant crop (barley, bermudagrass etc.) to provide ground cover for as much of the growing season as possible. If there is a white salty crust on the soil surface, delay planting the slat tolerant crop until the crust no longer forms during a soil drying cycle. Planting while the salty crust remains will likely result in poor stand establishment. The time for reclamation will depend upon the amount and quality of water that moves through the soil profile. Further information can be found in fact sheet 2226.

http://wpm9hr/soil/SalinityReport.asp?Login.LabID=249621 4/25/2001

Figure 27: A Web report with text box for county extension office



Page 1 of 1

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall, Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

WATER QUALITY REPORT

DARRYL CARTER

PO BOX 888
SULPHUR OK 73086
(580) 622-3181

Name:

Location:

Lab ID No.:	243618
Customer Code:	194
Sample No.:	4652
Received:	11/16/2000
Report Date:	4/25/2001

TEST RESULTS FOR *Irrigation Water*

-------- Cations -------		------ Anions ------		---------- Other-----------	
Sodium (ppm)	72	Nitrate-N (ppm)	1	pH	8.3
Calcium (ppm)	46	Chloride (ppm)	89	EC (µmhos/cm)	784
Magnesium (ppm)	25	Sulfate (ppm)	87	Boron	0.06
Potassium (ppm)	3	Carbonate (ppm)			
		Bicarbonate(ppm)	183		

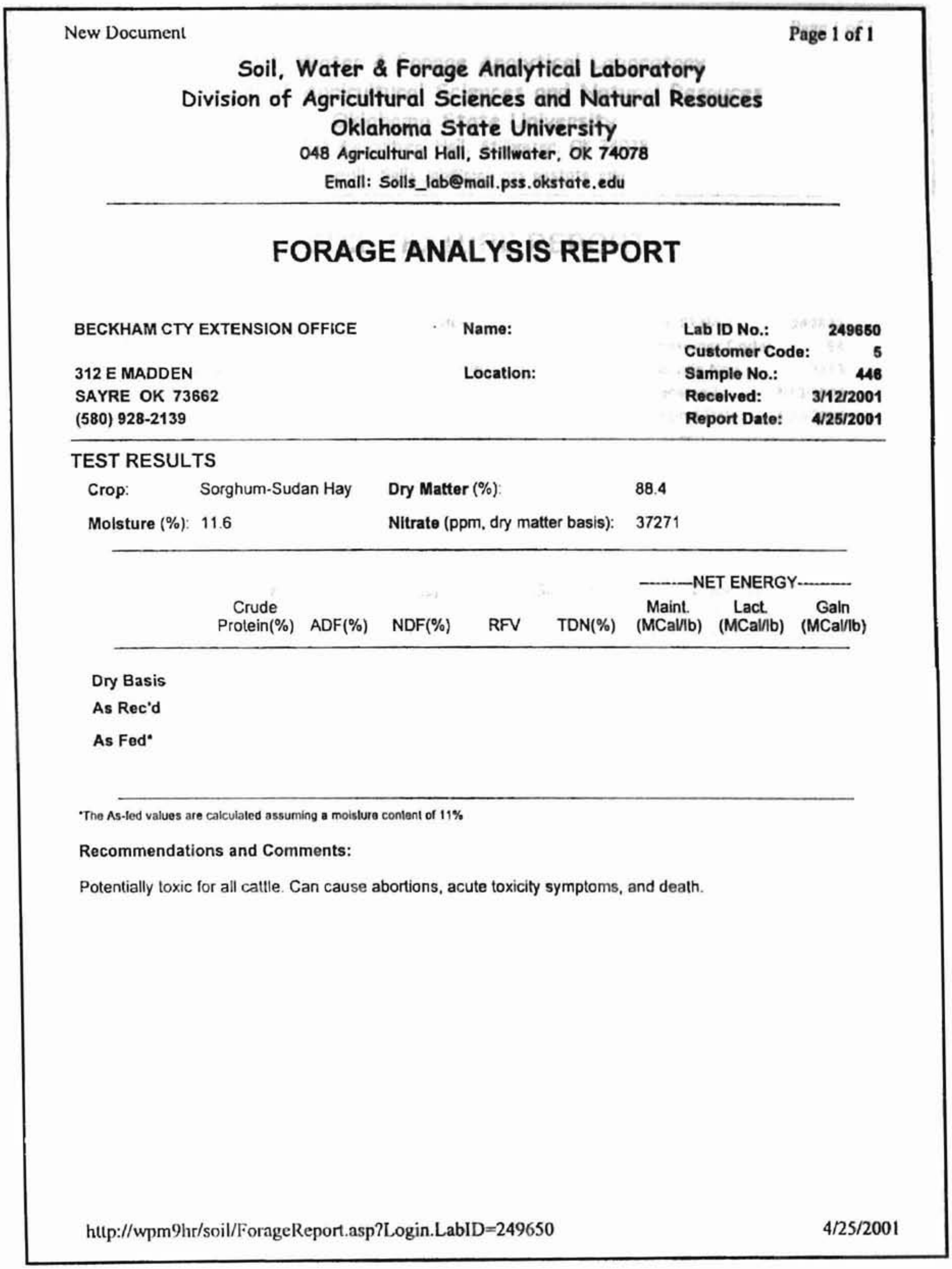
------------ Derived Values ------------		-------- Derived Values (cont'd) --------	
Total Soluble Salts (TSS in ppm)	527	Sodium Percentage	41.8%
Sodium ADsorption Ration (SAR)	2.1	Hardness (ppm)	218
Potassium Adsorption Ration (PAR)	0.1	Hardness Class	Very Hard
Residual Carbonates (meq)		Alkalinity (ppm as CaCO3)	150

INTERPRETATION AND REQUIREMENTS

County Extension Office Comments:

http://wpm9hr/soil/SalinityReport.asp?Login.LabID=243618 4/25/2001

Figure 28: A Web report for Forage test



New Document

Page 1 of 1

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall, Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

FORAGE ANALYSIS REPORT

BECKHAM CTY EXTENSION OFFICE	**Name:**	**Lab ID No.:** **249650**
		Customer Code: **5**
312 E MADDEN	**Location:**	**Sample No.:** **446**
SAYRE OK 73662		**Received:** **3/12/2001**
(580) 928-2139		**Report Date:** **4/25/2001**

TEST RESULTS

Crop:	Sorghum-Sudan Hay	**Dry Matter** (%):	88.4
Moisture (%):	11.6	**Nitrate** (ppm, dry matter basis):	37271

						---------NET ENERGY----------		
	Crude Protein(%)	ADF(%)	NDF(%)	RFV	TDN(%)	Maint. (MCal/lb)	Lact. (MCal/lb)	Gain (MCal/lb)
Dry Basis								
As Rec'd								
As Fed*								

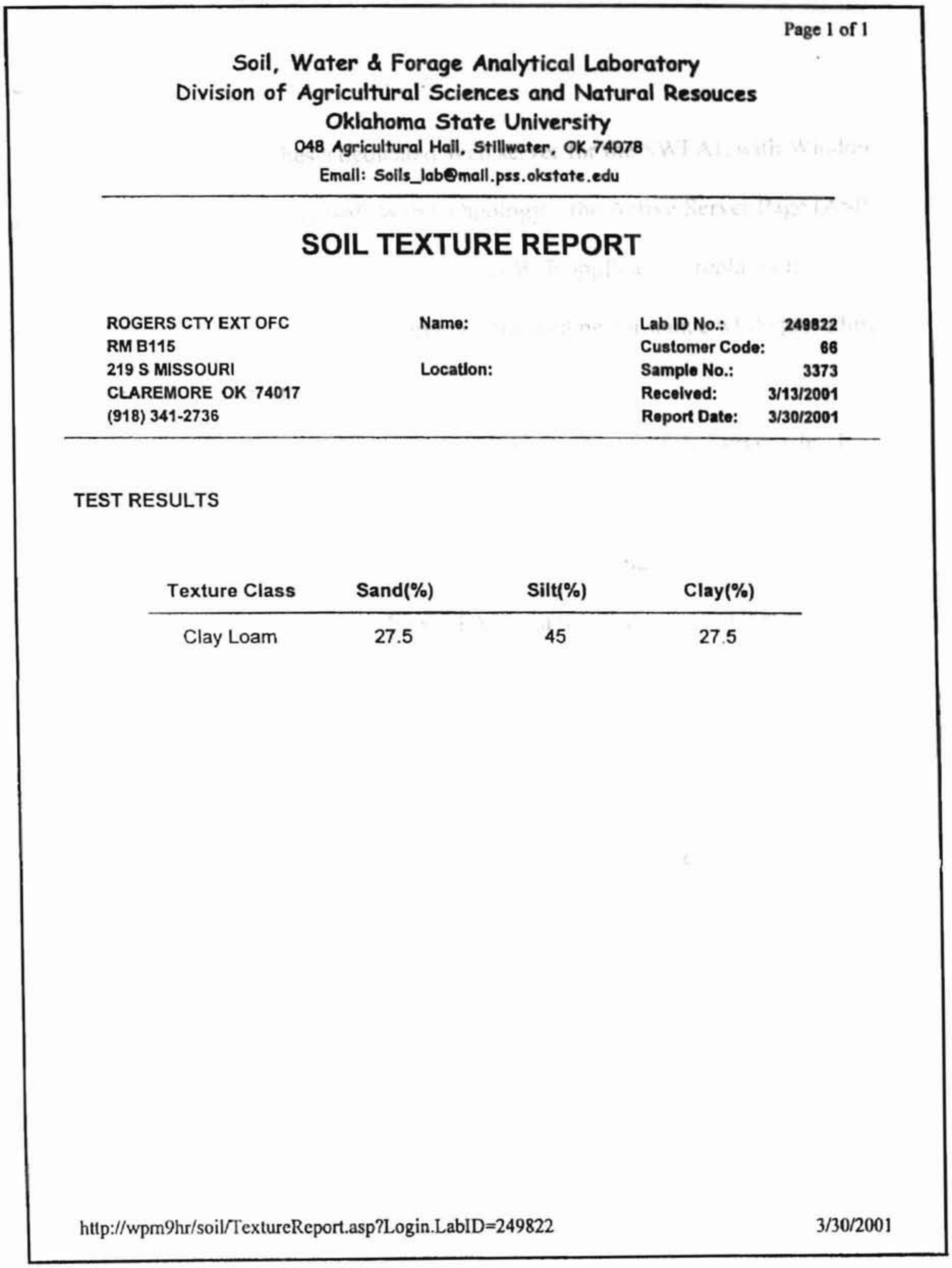
*The As-fed values are calculated assuming a moisture content of 11%

Recommendations and Comments:

Potentially toxic for all cattle. Can cause abortions, acute toxicity symptoms, and death.

http://wpm9hr/soil/ForageReport.asp?Login.LabID=249650 4/25/2001

Figure 29: A Web report for Texture test



Page 1 of 1

Soil, Water & Forage Analytical Laboratory
Division of Agricultural Sciences and Natural Resouces
Oklahoma State University
048 Agricultural Hall, Stillwater, OK 74078
Email: Soils_lab@mail.pss.okstate.edu

SOIL TEXTURE REPORT

ROGERS CTY EXT OFC
RM B115
219 S MISSOURI
CLAREMORE OK 74017
(918) 341-2736

Name:

Location:

Lab ID No.:	249822
Customer Code:	66
Sample No.:	3373
Received:	3/13/2001
Report Date:	3/30/2001

TEST RESULTS

Texture Class	Sand(%)	Silt(%)	Clay(%)
Clay Loam	27.5	45	27.5

http://wpm9hr/soil/TextureReport.asp?Login.LabID=249822 3/30/2001

V. SUMMARY AND FUTURE WORK

5.1 Summary

This study establishes a dedicated Web server for the SWFAL with Windows 2000 IIS 5.0, and it uses Microsoft Web technology – the Active Server Page (ASP) to realize dynamic Web report generating. This Web application replaces the SWFAL previous CGI Web program giving improved performance while providing Web visitors with better services.

The ASP program is written using VBScript and runs at the server site. It processes the users' requests, embeds the results into an HTML string then sends the results to each user's browser to format the resulted table and report.

With this Web application, the SWFAL customer may use any browser to view his/her own test results and to print test reports. The application eliminates data redundancy and possible data inconsistency. It guarantees the data presented on the Web report represent the most recent data in the database and protects the source code from being seen by the end user. When multiple users access the Web site at same time, the response time for each customer also improves.

The test report printed with this Web application may be used as a substitute for the hardcopy report, so the customers can get their reports much faster than previously. It is possible to save the SWFAL most of the cost for mailing test reports.

5.2 Future Work

When the SWFAL customer uses this Web application to print a test report, the user must first open the report page on his browser then print it from the report screen. If the customer, for example, a county extension office, has a large quantity of samples, then printing all of the reports requires opening a browser for each report. This is not an efficient way to print batch reports. To improve this process, a user side component is needed. When a user needs to print a batch report, the server may send the record set object to the user's computer, and the user side component can process the data received as well as format the reports and print them in a batch. This should make the process much faster.

Acronyms

ADF: Acid Detergent Fiber

ADO: ActiveX Data Object

ASP: Active Server Page

BI: Buffer Index

CERN: Conseil Europeen pour la Recherche Nucleaire

CGI: Common Gateway Interface

COM: Component Object Model

DBMS: Database Management System

DLL: Dynamic Link Library

DTD: Document Type Definition

EC: Electric Conductivity

ECC: calculated EC

EPP: Exchangeable Sodium Percentage

ESP: Exchangeable Potassium Percentage

GML: Generalized Markup Language

HTML: Hypertext Markup Language

IIS: Internet Information Server

IP: Internet Protocol

LAN: Local Area Network

NDF: Neutral Detergent Fiber

ODBC: Open DataBase Connectivity

OLE-DB: Object Linking and Embedding Database

OM: Organic Matter

PAR: Potassium Adsorption Ration

PERL: Practical Extraction and Report Language

pH: acidity

RAM: Random Access Memory

RDO: Remote Data Objects

RFV: Relative Feed Value

SAR: Sodium Adsorption Ration

SGML: Standard Generalized Markup Language

SWFAL: Soil, Water and Forage Analytical Laboratory

TCP: Transmission Control Protocol

TSS: Total Soluble Salts

UDA: Universal Data Access

URL: Uniform Resource Locator

WWW: World Wide Web

XML: eXtended Markup Language

Chemical elements, compounds

Al:	aluminum
B:	boron
Ca:	calcium
Cl:	chloride
CO3:	carbonate
Fe:	iron
HCO3:	bicarbonate
K:	potassium
Mg:	magnesium
Na:	sodium
NH4:	ammonium-N
NO3-N:	nitrogen
P:	phosphorus
SO4:	sulfate
Zn:	zinc

Bibliography

Aitken, P., 1996 Web Developer's Guide to JavaScript & VBScript, Coriolis Group Books, Scottsdale, AZ

Cailliau, R., 1999 Hypertext in the Web – a History, ACM Computing Surveys, Vol. 31, Number 4es

Chappell, D., 1996 Understanding ActiveX and OLE, Microsoft Press, Redmond, WA

Comer, D. E., 2000 The Internet Book, Prentice Hall, Upper Saddle River, NJ

Ding, M.J., 1999 Interconnection Between Web Server and DBMS Server, Computer_Application, Vol. 19, No.6

Homer, A., 1998 Professional ASP Technique for Webmasters, Wrox Press Ltd., Birmingham, UK

Labrinidis, A. and Roussopoulos, N., 2000 WebView Materialization, ACM SIGMOD International Conference on Management of Data (May)

Laurent, S. St., 1998 XML A Primer, MIS Press, New York

LeVitus, B. and Evans, J., 1996 Web Master, Academic Press, Inc., Chestnut Hill, MA

Ma, S. and Sheng, H. L., 1999 Research and Construction of a Database Middleware Prototype on Intranet, Journal of Software, Vo. 1, No.1

Marchal, B., 1999 XML By Example, Que, Upper Saddle River, NJ

Matzen, R. W., George, K. M. and Hedrick, G. E., 1997 A Formal Language Model for Parsing SGML, J. Systems Software; 36: 147-166

McCray, B. J., Jan. 2000 Poster Untitled, Soil, Water & Forage Analytical Laboratory, Oklahoma Cooperative Extension Service, Oklahoma State University, Stillwater, OK

Myers, B. et al., 1999 Past, Present, and Future of User Interface Software Tools, ACM Transaction on Computer-Human Interactions, Vol. 19, No.4

Nori, A. K., 2000 Application Architecture: 2Tier or 3 Tier? What is DBMS's Role? ACM SIGMOD, 1-58113-218-2/00/0005

Novalis, S., 1999 Access 2000 VBA Handbook, Sybex Inc., San Francisco, CA

Para, J., Brown, M., Caison, C., DeBetta, P., and Wilson, E., 1999 Professional ADO RDS Programming with ASP, Wrox Press Ltd., Birmingham, UK

Ramakrishnan, R., 2000 From Browsing to Interacting: DBMS Support for Responsive Websites, ACM SIGMOD, 5/00

Simon, A. R. and Wheeler, T., 1998 Open Client /Server Computing and Middleware, Academic Press Inc., Chestnut Hill, MA

Smith, N. E., 1998 Practical Guide to Intranet Client-Sever Applications Using the Web, Wordware Publishing, Inc., Plano, TX

Tittel, E., Madden, M., and Smith, D. B., 1996 Building Windows NT Web Servers, IDG Books Wold Wide, Inc., Danvers, MA

Townsend, J. J., et al., 1996 Web Development with Microsoft Resources, Sams.net, Indianapolis, IN

Ullman, C., et al., 1999 Beginning ASP 3.0, Wrox Press Ltd., Birmingham, UK

Viescas, J., 1999 Running Microsoft Access 2000, Microsoft Press, Redmond, WA

Wang, G. H. and Shu, Y. T., 1999 Developing Dynamic and Interactive Web Pages by ASP, Computer Applications, Vo.19, No.5

Wang, Y. Y. and Jin, F., 1999 Comparison with the Technical Property of CGI and ISAPI, Computer Applications, Vol. 19, No.4

Wehling, J., et al., 1996 Webmaster's Ultimate Resource Guide, Ziff-Davis Press, Emeryville, CA

Weissinger, K. A., 2000 ASP In a Nutshell, O'Reilly, Sebastopol, CA

2

VITA

Duan Sheng

Candidate for the Degree of

Master of Science

Thesis: Dynamic Web Publishing – Analytical Report

Major Field: Computer Science

Biographical:

Education: Received Bachelor of Science degree in International Trade and Economics from Jiangsu Foreign Trade Institute, Nanjing, China in July 1985. Completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University in June, 2001.

Experience: Employed by Jiangsu Light Industrial Products Imp. & Exp. Company as a sales manager, from January 1981 to January 1991; employed by Oklahoma State University, Department of Computer Science as a graduate teaching assistant, from August 1999 to December 2000; employed by Oklahoma State University, Soil Water & Forage Analytical Laboratory as an information system and Web application developer from May 2000 to present.