

A WEB-BASED INFORMATION TECHNOLOGY  
SOLUTION FOR A SMALL BUSINESS

By

SARAH CHERIYAN

BACHELOR OF ENGINEERING

UNIVERSITY OF KERALA

KERALA, INDIA

1995

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  
December, 2001

A WEB-BASED INFORMATION TECHNOLOGY  
SOLUTION FOR A SMALL BUSINESS

Thesis Approved:

*D. E. Howell*

Thesis Adviser

*Blayne E. May*

*Blayne E. May*

*Timothy J. Pittman*

Dean of the Graduate College

## ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor Dr. G. E. Hedrick for his guidance and supervision for the completion of my thesis work. I would also like to express my deep sense of gratitude to Dr Chandler, Dr Mayfield and Dr Dai for serving on my graduate committee and for providing valuable suggestions and ideas.

I am also grateful to the founder and President of Geneseek, Inc and his staff for all the technical support and ideas that were critical for the completion of this work.

My greatest appreciation goes to my family and friends for their loving support and inspiration and above all to my Lord for having given me his protection and wisdom.

## TABLE OF CONTENTS

Chapter	Page
1. INTRODUCTION.....	1
1.1 Electronic Commerce on the Internet.....	1
1.2 Background.....	2
1.3 Purpose of the Project.....	3
1.4 Objectives of this Project.....	4
2. REVIEW OF LITERATURE.....	7
2.1 Dynamic Web Applications.....	7
2.2 RDBMS for a Client/Server Environment.....	8
2.3 Security Solutions for the Enterprise Infrastructure.....	11
3. A WEB BASED BUSINESS SOLUTION.....	20
3.1 Description of Applications.....	21
3.2 Design and Implementation of Database Tables.....	41
4. SECURE REMOTE ACCESS INFRASTRUCTURE.....	44
4.1 Internet Security Protocol Alternatives.....	46
4.2 Authentication and Authorization of Users.....	50
4.3 Message Encryption and Authentication.....	52
4.4 Network Address Translation.....	52
4.5 Firewall Service.....	53
5. SUMMARY AND CONCLUSION.....	55
REFERENCES.....	57

## LIST OF FIGURES

Figure	Page
2.1 An Interactive Business Application.....	8
2.2 SSL Handshake Process.....	13
2.3 IP Packet Headers.....	16
3.1 Sequence of Operations in a Web Application.....	21
3.2 Registration Form.....	23
3.3 Registration Confirmation.....	23
3.4 Order Form.....	26
3.5 Order Detail Form.....	28
3.6 Billing Statement.....	29
3.7 Select Records for Posting Arrival .....	31
3.8 Update Arrival .....	31
3.9 Payment Form.....	33
3.10 Account Information.....	35
3.11 Schedule Samples for Testing.....	37
3.12 Save Results to Databases.....	39
3.13 Retrieve Records of Results.....	40
3.14 Test Results.....	41
3.15 Relationship between Database Tables.....	43

## NOMENCLATURE

AH:	Authentication Header
CGI:	Common Gateway Interface
CPU:	Central Processing Unit
DNA:	Deoxyribo Nucleic Acid
DRI:	Declarative Referential Integrity
ESP:	Encapsulating Security Protocol
HTML:	Hyper Text Markup Language
HTTP:	Hyper Text Transfer Protocol
IKE:	Internet Key Exchange
IP:	Internet Protocol
IPSec:	Internet Protocol Security
ISAKMP:	Internet Security Association and Key Management Protocol
ISQL:	Interactive Standard Query Language
LDAP:	Light Weight Directory Access Protocol
MAC:	Message Authentication Code
PKI:	Public Key Infrastructure
RADIUS:	Remote Authentication Dial In User Service
RDBMS:	Relational Database Management System
RI:	Referential Integrity

SA: Security Association  
SADB: Security Association Database  
SQL: Standard Query Language  
SSL: Secure Socket Layer  
URL: Uniform Resource Locator  
VPN: Virtual Private Network  
WWW: World Wide Web

## CHAPTER 1

### INTRODUCTION

#### 1.1 Electronic Commerce on the Internet

The World Wide Web has elevated business communication to new heights in the last decade. Internet protocol routing coupled with optical networks can create high speed access to massive amounts of information residing on super computers. Thus web usage has grown beyond browsing and text search to include new functions such as seamless cross platform exchange of data with different semantics, and intelligent interactive applications for information processing. Companies have often made the mistake of underestimating the role of the web in electronic commerce as a mere marketing tool to attract customers. Instead the failures in electronic business ventures teach us that it is a vital transformation needed for improving business collaborations and business transactions with partners, customers, and employees. Companies should commit themselves to an aggressive strategy for an electronic business infrastructure at the network edge to automate customer-to-business, employee-to-business, and business-to-business transactions with its databases. It should keep up with the latest advances in network technology to ensure a secure and smooth access to its resources.

A newcomer to electronic commerce is overwhelmed by the problem of streamlining his business operations for an Internet market. It begins by developing customer friendly software solutions to automate business interaction



with customers, employees and business partners. Application servers and database servers are needed to host the applications and back office system involved in these operations. In addition network perimeter devices such as authentication servers and access control directories are needed to monitor, and authorize traffic. An assessment of the risk involved with different classes of transactions will determine the security protocols and monitoring tools necessary to authorize, authenticate and secure traffic.

## **1.2 Background**

The company, Geneseek Inc., is a biotechnology company based in Lincoln, Nebraska. It provides special research services to organizations like pharmaceutical companies, biotechnology ventures, and universities. Their gene discovery and DNA (deoxyribonucleic acid) analysis services include automated DNA sequencing, genotyping, and fragment analysis. The company currently has a web presence in the form of static pages covering product description and company profile. In order to establish a competitive advantage the company must establish a reliable and functional web interface to automate its business-to-customer and business-to-business communication needs. In this process they aspire to build the world's first "virtual laboratory" on the Internet. Given the past history of insecurity and best effort service until a few years ago, the Internet could not guarantee a secure exchange of DNA information. Internet privacy and secrecy is a relatively new concept that has been gaining momentum in the past few years and protocols are constantly evolving to keep up with the demands of the market. A new venture is not only faced with the task of accurately assessing the security risks and reviewing the best counter measures

available in the market, but also expected to constantly review and update technology to meet the current and future demands by partners for direct access privilege to internal data systems using directory services. As the business expands the infrastructure should step up to support services such as applications that allow cross platform exchange between proprietary applications, as well as applications that allow direct directory-to-directory interaction [18].

### **1.3 Purpose of the Project**

The company, Geneseek Inc., needs assistance in identifying an effective strategy for its migration to the web. This project focuses primarily on designing and implementing a working model of a back office system to manage customer information and customer related operations of the workflow such as accounting, inventory, and order logs, etc. Relevant web applications are created to satisfy the needs of customers, employees, and partners who wish to interact with the back office system. System and network advances necessary to handle the added influx of business generated by the new business opportunities are determined. In addition the security concerns of partners, customers and the company has been reviewed and the latest alternatives evaluated to recommend a reliable mechanism to monitor access points to the network, authorize and encrypt sensitive transactions. Interoperability of protocol implementations at remote sites and the central office is a major consideration for selecting communication protocols and network security protocols for connecting multiple remote networks to the corporate office. Another problem is that there are too many new products in the market with little or no field-testing supporting their claims. The best way to follow through with a transition strategy under such circumstances is to

reassess alternatives and readjust the transition strategy to accommodate setbacks and new demands. The fundamental purpose of this project is to assess the alternate technologies and recommend the ones that significantly enhance web operations for Geneseek.

## **1.4 Objectives of this Project**

### **1.4.1 Develop and Implement Interactive Web Applications**

There are few web tools for developing business applications for a virtual laboratory. This process is complex and requires a thorough understanding of workflow. The main objective is to create customized web applications that mediate dynamic interaction between a corporate database and the customers and employees at remote locations. Customers with standard access privileges are able to register as a new user, enter orders, and sample descriptions from a remote location. They can track payments, order status, account history, and balance information from the convenience of their homes using an Internet connection. Additional applications are accessible to privileged customers who adopt special security measures at remote locations. In addition a set of applications are available for employees on location or at remote locations to access relevant databases to modify and update information. Geneseek should also re-engineer its back office system to host its web applications. This new function requires a separate relational database to manage client related information. The database is designed to store client information, order logs, sample details, payment information, and account balance. Web applications act as a front end to the database. It consists of web pages, and Common Gateway

Interface (CGI) programs that trigger database queries and procedures to modify or query database information on the fly.

The company prefers to maintain a vendor independent infrastructure for its web business. In keeping with the financial limitations of a new venture they want to make the best sense of their investment. By adopting Linux as the network operating system, and associated Internet software's such as Apache Server, and Sybase Adaptive Server, it can achieve its goals at an affordable cost [11].

#### **1.4.2 Enterprise Infrastructure for Secure Web Operations**

Definitions such as “virtual laboratory”, “virtual enterprise”, and “knowledge networks” are relatively new concepts for which there are no set standards on the Internet. Application developers that launch these functions want to use the Internet as a medium of communication between computer systems, and people engaged in data intensive tasks and collaborations [18].

The Internet, as such, lacks intelligence and has no guarantees for secrecy and authenticity of transaction. Security measures are imposed at the network perimeter to authorize access to clients and employees on the basis of an authentication mechanism that proves the identity of the remote host. Because of the sensitive nature of laboratory records, additional measures are needed to encrypt sensitive files before it leaves the corporate premises.

In order to qualify as a virtual laboratory, and deliver on the promises made to the biotechnology community it is imperative that the company's web infrastructure be designed to incorporate state of the art technology to meet the following service expectations:

- Enable multiple secure remote sessions over the Internet for limited access to customer's personal information databases such as inventory, order logs and account information for the sake of automating customer related operations. Such sessions require mutual authorization of clients involved in a communication. In addition data should be exchanged securely by using a suitable encryption algorithm.
- Enable secure read access for partners to specific databases that store test results and other information.
- Enable secure remote access over the Internet for employees who wish to manage operations from home.
- Identify a suitable mechanism for managing client authorization using passwords or public keys.
- Identify suitable precautionary measures and monitoring devices for strengthening the network perimeter to minimize the occurrence of vulnerabilities and block unauthorized traffic.

## CHAPTER 2

### REVIEW OF LITERATURE



#### 2.1 Dynamic Web Applications

The use of Common Gateway Interface (CGI) programs enables web servers to generate web pages with dynamic web content. They allow web servers to customize server response to suit the specific nature of enquiries from remote clients. This new feature attracted enterprises to offer unique business services such as customer driven applications that answer customer queries with up-to-date information from system servers and databases [1]. Gundavaran describes the various steps involved in building interactive web applications by solving some practical problems [1]. He describes how CGI programs can be formulated to generate forms for accepting questions from users, and how it operates as a web gateway by executing SQL queries on behalf of remote clients to access information stored on information servers such as Usenet, Oracle, and Sybase [1]. He then outlines how the data read from the servers can be reformatted, and modified with appropriate HTML tags to prepare a virtual document for the client. Figure (2.1) describes the steps involved in a typical web gateway application.

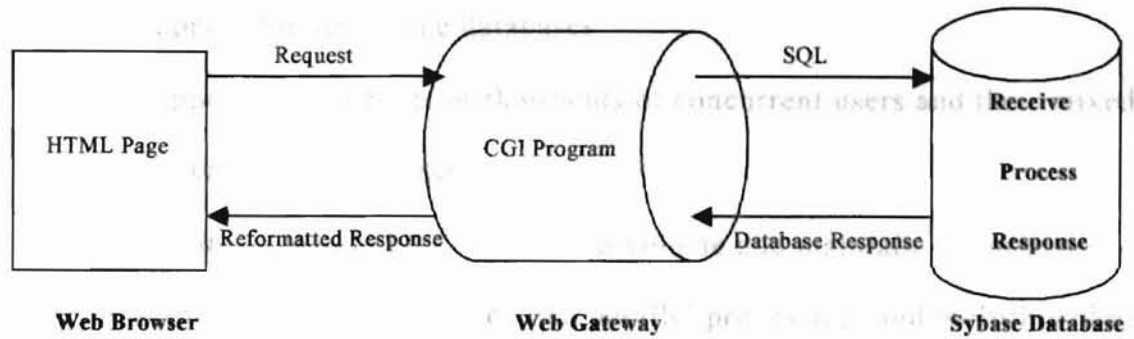


Figure (2.1)  
An Interactive Business Application  
Adapted from [1]

Guidelines from his book were used to develop web gateway applications to allow customers to post orders, ask questions about accounts and payments made, and access files that store test results from database. Another group of applications were generated for the employees to manage certain business operations from remote site.

## 2.2 RDBMS for a Client/Server Environment

Sybase Relational Database Management System (RDBMS) competes in a commercial market that places high expectations on a distributed client server environment for managing information resources. The Sybase database system has improved over the years to keep up with the exponential growth in client population, size of databases, and volume of transactions. Anderson describes some new features in Sybase 11 that has earned a reputation for reliability and high performance in a transaction intensive business environment [2].

The key techniques include [2]:

- Online Transaction Processing (OLTP)
- Decision Support Systems (DSS) to add intelligence to the server

- Support for very large databases Database Design
- Support for hundreds or thousands of concurrent users and their mixed as workload of transactions . . . . . is a standard naming
- Ability to handle high transaction volume and maintain performance standards using techniques like parallel processing, and multithreading.
- Scalability

Sybase RDBMS fits the requirements of a typical web based business environment so it is a suitable choice for managing information for Geneseek's back office system.

Performance is at the core of any database implementation, so it is important to maintain the same priority in designing the logical concepts for the database, and the physical layout of the database. Sybase has some efficient programming features such as control-of-flow, triggers, stored procedures, rules, and defaults on its latest Transact SQL server [2]. Additional performance benefits are achieved by identifying the most appropriate layout for the tables so that data necessary for transactions can be accessed with fewer queries and input/output operations. Anderson suggests the use of index to identify records, clustered index to maximize cache usage, normalized tables to avoid repetition of attributes in multiple tables, and SQL join statements to combine data on related tables [2].



### **2.2.1 Concepts and Terms Relevant to Database Design**

User databases, database tables, procedures, views, and rules are identified as objects within the server. *database.owner.objectname* is a standard naming format for database objects.

#### **2.2.1.1 Indexing**

Indexes are data structures that store the record key and the address of the table records. Indexes are arranged in ascending order of the keys and the records are arranged in the same order as the keys. Indexes help in reducing time needed for search operations, and maximizing the efficiency of a storage cache. Column names in tables are clustered based on the order in which application procedures access records from a table [2].

#### **2.2.1.2 Referential Integrity**

SQL statements can join tables only if they have a one-to-many relationship. This relationship is enforced by Referential Integrity (RI) constraints that prevent updates or changes to the database tables that violate the relationship. Declarative Referential Integrity (DRI) is the explicit declaration of a primary key and foreign key when tables are first created [2]. RI can also be enforced using procedures called triggers. Triggers enlist the rules needed to maintain RI when related tables are updated using INSERT, DELETE, and UPDATE commands. It executes a ROLLBACK TRANSACTION command if database updates do not comply with the RI rules [2].

### **2.2.1.3 Defaults and Rules**

These objects ensure that only acceptable values are assigned to columns. A default value is assigned to a column automatically if no value is specified for the column in an INSERT command. Rules ensure that values assigned to a column abide by the restrictions imposed on the column [2].

### **2.2.1.4 Interactive SQL**

Business applications use an Interactive SQL (ISQL) interface to interact with the SQL server. CGI programs create executable files that contain commands to initiate an ISQL interface. The executable file then passes SQL commands to the ISQL interface. The CGI program reads the response from the interface, modifies the results with appropriate HTML tags, and displays the results on a web page [2].

## **2.3 Security Solutions for the Enterprise Infrastructure**

The company, Geneseek Inc, sees the enormous risk of exposing its critical internal infrastructure to the Internet. Most security breaches occur due to the lack of a strict security policy. Even the best technology is ineffective unless good security habits are practiced. There are several aspects to security including database security, network security, and remote access security. The primary focus is on identifying suitable techniques for enforcing security for the network perimeter to contain unauthorized intrusion from sources outside the corporate network. Some of the techniques discussed involve security solutions such as Secure Socket Layer (SSL) protocols for ensuring secure remote access, Internet Protocol Security (IPSec) protocol for secure high speed access, Firewalls for blocking

unauthorized traffic, Network Address Translators (NAT) for concealing network information from hackers, Public Key Cryptography for sharing keys in secret.

The underlying operating system that ties the different network functions plays a major role in strengthening the network perimeter. Linux, a unix variant for desktop architecture is extremely reliable and inherently resistant to many viruses [11]. There is an added financial advantage to this operating system because many of the key security solutions discussed in the following sections are available as freeware on Linux. More vendors are supporting Linux on their latest products in light of the growing interest in this operating system. Consequently network choices are governed to a great extent by the functions and services they support and the operating system that connects them to each other. Some network security alternatives are discussed in the following sections.

### **2.3.1 Secure Socket Layer Solution**

When people share financial information or any form of sensitive information they need to confirm the identity of the person with whom they are communicating, know for certain that the information was not tampered during transit, and messages remains a secret until it reaches the other person. Secure Socket Layer (SSL) protocols add these security attributes to data above the transport layer on the computer where the data originates, to ensure a safe context sensitive communication from computer-to-computer [5]. SSL is increasingly popular and affordable because SSL client implementation is preinstalled on the Netscape browser on almost all windows based personal computers, and on the server side SSL module is a freeware for apache web server [4]. SSL handshake protocol and SSL record protocol are the two

components of the SSL protocol suite. SSL handshake protocol is responsible for initiating a session between two entities by authenticating the peers and negotiating a mutually agreeable encryption algorithm, message authentication algorithm and its respective keys [3]. At the end of a handshake protocol a secure session is established and record protocols step in to encrypt the actual messages and maintain its integrity. Figure (2.2) shows the SSL handshake procedure.

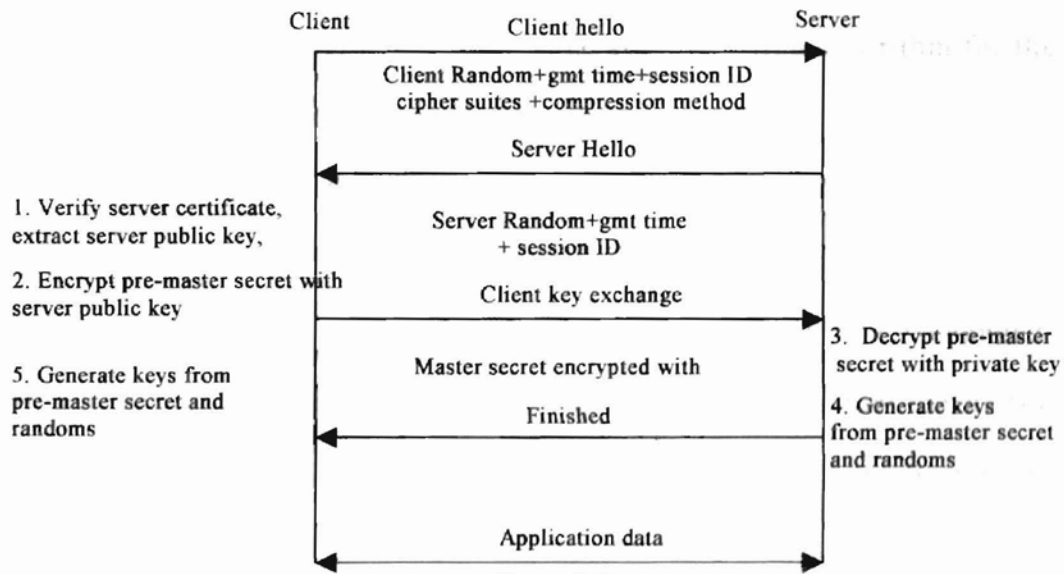


Figure (2.2)  
SSL Handshake Process  
Adapted from [3]

SSL supports popular public key cryptography algorithms to authenticate the corporate server and the client to each other. A pair of keys namely a private key and a public key is assigned to each client [3]. A Corporate server publishes its public key to its customers/clients on a certificate and keeps its private key a secret. The client verifies the certificate and public key issued in favor of the corporate server by a certification authority [3, 4]. The same process is repeated to authenticate the client to the corporate server. Once the clients are

authenticated they proceed to share a secret. A client encrypts a secret using the public key of the corporate server. On receiving an encrypted message the server decrypts it using its private key. From now on the server and the client have a shared secret. Encryption and decryption using public key algorithms involve modular exponentiation to a very large base. Hence it is computationally intensive and unsuitable for the actual message [3]. A more efficient alternative is to use public key algorithm for the initial handshake and a symmetric algorithm for the actual message. During handshake clients agree on a mutually acceptable symmetric algorithm, a message authentication algorithm for the message, and symmetric keys needed for the two algorithms. One time symmetric keys have a short life span and hence it keeps the hacker from cracking the key [4, 5]. Record protocols use the symmetric algorithms and the session keys negotiated by the handshake protocols to encrypt the actual text exchanged during a session. In order to preserve the integrity of the content during transit and authenticate the source of the message, a Message Authentication Code (MAC) is created using a message authentication algorithm and a shared secret key [3]. A client who receives the message verifies the integrity by passing the message through the same MAC algorithm and then keying the same shared secret. Integrity of the message is guaranteed by SSL, if the MAC generated at the receiving end matches the original MAC [5].

### **2.3.2 Virtual Private Networking using IPSec Protocol**

Virtual Private Networking (VPN) is a relatively new secure data transmission mechanism that is a popular choice for connecting small and mid size corporations to its partners and clients. It has enormous business potential

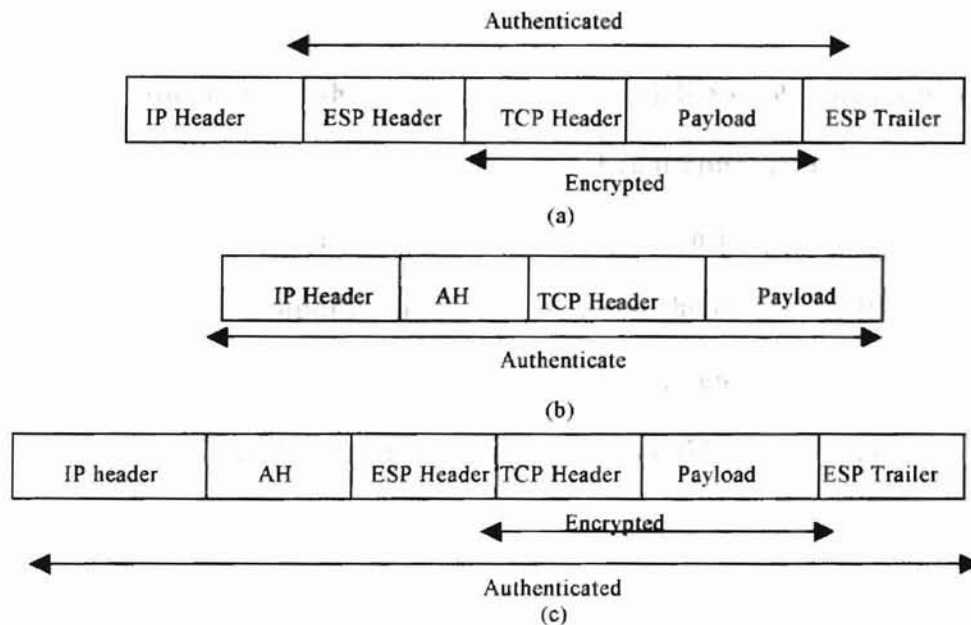
because it combines IP security protocols with broadband public networking resulting in remote access connections comparable to a private line connection at a fraction of the cost [8].

Two types of VPN implementations are possible: client implementation at each remote site, and a VPN gateway implementation at the corporate center. Most of the VPN implementations follow IPSec protocol specifications for ensuring that nodes involved in a communication are authorized, data exchanged is not altered, and data remains a secret during transit [5, 8]. IPSec is implemented at the network layer so all Internet Protocol (IP) based traffic and any other traffic that travels over IP can use the VPN service [5]. Compared to SSL, it is a more sophisticated security solution in terms of enhanced performance in a distributed access environment, ability to establish consistent security standards irrespective of the diversity of the environments connected, and remarkable scalability to adapt to a high speed shared broadband environment [6].

IPSec suite has a robust mechanism that manages security on a per flow basis allowing clients to negotiate different levels of security by choosing some or all of the security attributes such as authentication, encryption, or both over individual connections [9]. Authentication Header (AH) protocol and Encapsulation Security Protocol (ESP) are the two protocol components of the IPSec suite that determine the level of security attributed to a connection.

These protocols encapsulate IP data packets using an AH or ESP packet header or in some cases both AH and ESP headers before they are transported over the Internet [9]. Figure (2.3) shows the packet encapsulations for IP traffic [9]. The port receiving the packets use the information on the AH header encapsulating IP packet to

verify proof of origin of packets, data integrity, and anti replay detection. ESP header on the other hand does all that an AH does in addition it ensures data confidentiality and traffic flow confidentiality by encrypting the content of the packets [5].



(a) IP Packet with ESP applied (b) IP Packet with AH applied (c) IP packet with ESP and AH applied

Figure (2.3)  
IP Packet Headers  
Adapted from [9]

IPSec protocols exercise separate security policies for individual traffic flows. If two nodes A and B are communicating using IPSec then node A maintains an IPSec policy called Inbound Security Associations ( $SA_{in}$ ) for traffic flow from B and another policy called outbound Security Association ( $SA_{out}$ ) for traffic from A to B [5]. An SA is protocols specific, hence if the two nodes use AH and ESP protocols then each node stores two pairs of SA's [5]. An SA policy describes the security arrangements between the two nodes. An IPSec implementation reads the SA policy from an SA database to determine the security measures exercised by the AH header or ESP header for traffic bound

for a destination [9,5]. When a node initiates a connection for the first time there is no SA associated with the destination address. IPsec uses a handshake protocol called Internet Key Exchange (IKE) protocol to mutually authenticate the peers and negotiate the specific details of the SA such as algorithm and shared secret keys.

IKE protocol stores this information as an IKE SA. Whenever a session is set up the SA database is referred for an SA for that connection. If SA does not exist it refers to an IKE SA specification to negotiate an IPsec SA for that session [5, 9]. IPsec SAs are temporary and they expire at the end of its lifetime. Using the information on IKE SA, an AH SA or ESP SA is generated repeatedly when SA's expire [9, 10]. The keys for generating an IPsec SA are derived from the IKE SA.

### **2.3.3 Access Control**

Role based access control is a necessity in a distributed environment served by a VPN. This is because VPN serves as a communication channel for two or more classes of service namely customer-to-business, employee-to-business, and business-to-business. The risks are high if authorization and access privileges are the same for all clients [16]. Role Based Access Control is implemented using a Lightweight Directory Access Protocol (LDAP) that acts as a mediator between a user who requests access to resources and the central repository that stores access policies of all resources [5]. This central policy storage is complex and requires an excellent mediation system to maintain communication and synchronization between various protocols, clients and servers. Gutzmann describes the role of LDAP in translating policy from a higher level to a format more appropriate for network layer based security protocols such as



IKE, AH and ESP [16]. The policy description for an IPSec based VPN networks is very complicated. Doraswamy describes the attributes of an IPSec compliant policy [5].

#### **2.3.4 Authentication Mechanisms used by Internet Security Protocols**

Authentication is an important phase in verifying the credibility of clients that use the corporate resources. IP address is a weak form of authentication because a hacker can easily impersonate an IP address and earn access to the corporate network. Perlman and Kaufman describes authentication schemes based on public keys and pre-shared keys [10]. The services provided by each mechanism, and its effectiveness in actual business scenarios are also reviewed.

#### **2.3.5 Network Address Translation**

Network Address Translation (NAT) is a necessity in large IP based web environments. Under the current conditions IP protocols are unable to deliver exceptional web services because of the scarcity of registered IP addresses. NAT solves this problem to some extent when they are correctly configured in a business environment [12]. Liebmann discusses the network scenarios that are most suited for a NAT solution. He also studies the direct contradiction that some protocols face in interoperating with a NAT enabled network [12]. Shieh et al evaluates the root cause of these issues and determines a list of protocols that are likely to be affected by a NAT [13].

### **2.3.6 Firewall**

A Firewall is an integral part of network security. The firewall functions have increased over the years. Current VPN solutions can implement a firewall policy at the router or switch without causing serious back up in traffic. Firewalls of this kind usually monitor traffic at the network level. Firewalls are typically installed at the remote location as well as the corporate site for serving specific purposes. Blancharski describes the nature of functions performed by firewalls at the network layer [20]. King describes the security risks associated with VPN implementations at the client site and suggests the use of personal firewall to eliminate some of the vulnerabilities [6]. In another article he discusses the extended functions performed by firewalls for monitoring irregularities in traffic [19].

## CHAPTER 3

### A WEB BASED BUSINESS SOLUTION

A web application is a tool that allows a remote user to interact with the business logic residing on a corporate network. A web page is the first point of contact for all classes of users. With the help of a simple visual presentation these pages convey the purpose of each application. Forms use appropriate text fields to accept information necessary to answer specific questions, or to perform specific functions for the client. The data submitted through forms are sent to the web server in Hyper Text Transfer Protocol (HTTP) format. The intelligence of an application resides in the executable Common Gateway Interface (CGI) programs on the web server. It reads data from the form, converts it into a format acceptable to the application, and generates SQL queries for accessing the database to store, to retrieve, or to update information. SQL queries are embedded in a script file that is executed by the ISQL interface. It then awaits response from the ISQL interface and displays the response back to the user in a web page format. A user who initiates the process is unaware of the sequence of operations that take place in the background. Forms that initiate the applications can be accessed from a web browser on the client's machine. The CGI programs that interpret the contents of the form are written in the C programming

language, precompiled, and stored in the “/cgi-bin/” directory on the web server.

Fig

ure (3.1) describes the steps that take place behind the scene in a dynamic web application [1].

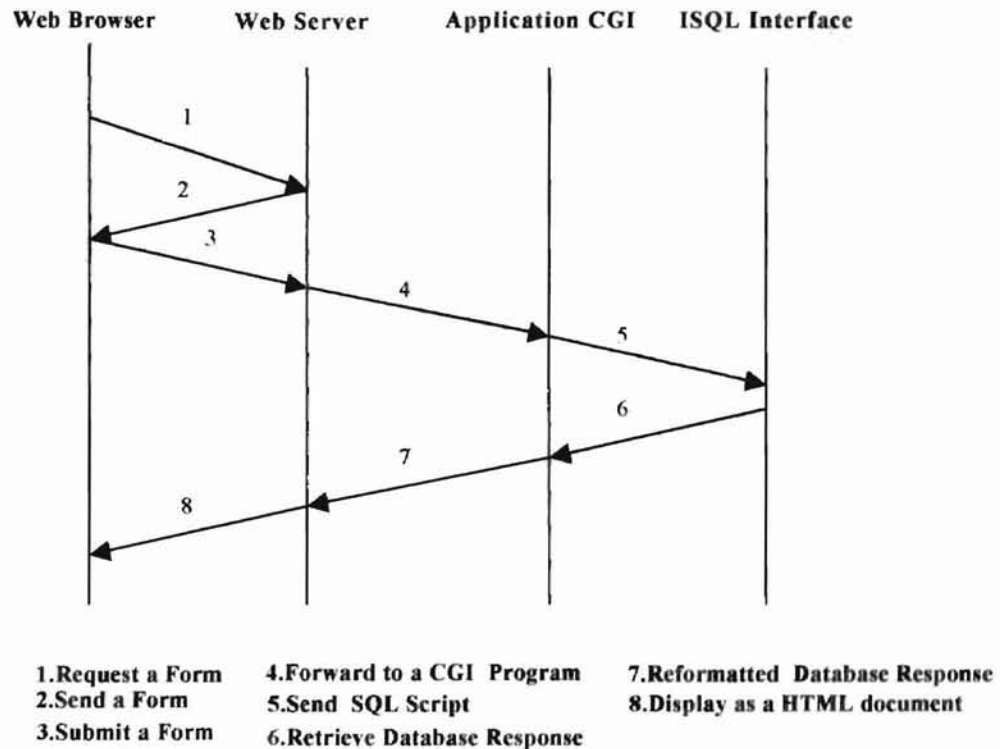


Figure (3.1)  
Sequence of Operations in a Web  
Application  
Adapted from [1]

### 3.1 Description of Applications

Each application description includes a brief outline of the CGI executable files, and the SQL procedure necessary to complete the transaction. The names of the CGI programs, SQL procedures, and database tables are highlighted in bold lettering. A visual presentation of the sequence of interactive HTML pages that

illustrate the operation sequence are also included. Due to large size of the source files, a soft copy is available for further reference.

### **3.1.1 Registration of New Users**

This process is done to open new accounts for customers in the database table **customers**. When the registration is completed users are identified by a unique identity *customers.customerid*. All other applications use this unique identity to track information about the customer from other related database tables. Figure (3.2) shows a registration form for accepting information from new customers. Information from the form is submitted to a CGI program. Figure (3.3) shows the confirmation displayed on the monitor when registration is complete.

#### **3.1.1.1 customerregistration.cgi**

This CGI program performs necessary steps to convert data from HTTP format to text format. It then creates a unique identity for the customer by attaching a random number to the last name of the customer. It then executes a Sybase procedure **scanforcustomer**. This SQL procedure first scans the customer's table to verify the presence of any previous records. If the search returns a record then the customer is notified that he is already registered and he is allowed to proceed to avail other web services, if the search returns no records a new row is inserted into the database table **customers**.

**Fill out your personal information and SUBMIT the completed form**

**Registration Form**

<b>FirstName</b> JOHN	<b>LastName</b> DOE
<b>Title</b> SCIENTIST	<b>Department</b> VETMED
	<b>Company</b> UCLA
<b>Street Address</b> 1344 N WILLOW AVE	<b>City</b> BROKENARROW
<b>State</b> Alabama	<b>ZipCode</b> 74012
	<b>Country</b> USA
<b>Phone-Number</b> (918) 249-2082	<b>Extension</b> 1
<b>Email Address</b> DOE@YAKOO.COM	<b>Fax Number</b> (939) 999-9999

submit\_registration    Reset

Figure (3.2)  
Registration Form

**Your registration was successful, please avail our services when you receive your new password and userid**

**Thank you JOHN DOE, for visiting our site**

[Geneseek Home Page](#)

Figure (3.3)  
Registration Confirmation

When registration is complete the new customer is sent an email notifying him of his new user identity. This email enters an inbox on a local email server, and emails are sent out the following business day.

**3.1.1.2 scanforcustomer ( @companyname, @firstname, @ lastname, @ department, @ address, @ city, @ state, @ zipcode, @ country, @ title, @ phone, @ extension ,@ fax, @ userid , @ email )**

This SQL procedure receives its parameters from the CGI program. Using the parameters *@companyname*, *@lastname*, and *@firstname* it verifies if there are any matching records in the database table **customers**. If the SQL select statement returns no matches then it uses an insert statement to create a new customer account. This procedure returns a confirmation back to the CGI program when the record is posted successfully.

### **3.1.2 Accept Orders Online**

Customers and partners use an order form to place orders electronically. An order form accepts information such as user name, customerid, shipping address, type of mailing, and type of payment, ...etc from the customer. A CGI program processes information from the order form. On successfully completing an order, the information is saved to the database table **ordertable** and then user is directed to an order detail form. Figure (3.4) displays the form for accepting the order and the sampledetails.

### **3.1.2.1 processorder.cgi**

The data received from the order form is converted from HTTP format to text format. An order record is identified by a unique identity *ordertable.orderid*. A random number is generated and attached to the customer's surname to create the identity. It then executes an SQL procedure **saveorder** to save information to the database table **ordertable**. When the record is saved successfully the CGI program generates an order detail form for the customer.

**3.1.2.2 saveorder( @orderid varchar(15), @shipfirstname varchar(30), @shiplastname varchar(30), @customerid varchar(20), @rdate varchar(20), @address varchar(100), @city varchar(30), @state varchar(30), @zipcode varchar(30), @country varchar(30), @phone varchar(30), @ponumber varchar(10), @paymenttype varchar(30), @shipmethod varchar(30) )**

This SQL procedure receives its parameters from a CGI program **processorder.cgi**. Based on the method of shipping *@shipmethod*, the freight charge is determined for the order. It uses the Sybase function `getdate ( )` to determine the date on which the order was made. The parameters are posted as a new record to the database table **ordertable**.



The screenshot shows a Netscape browser window titled "Netscape: Order Entry Form". The address bar contains "http://localhost/geneseek/order\_form.html". The page content is titled "ENTER YOUR ORDER INFORMATION HERE" and contains the following form fields:

FirstName	LastName	Customerid	Required Date mm/dd/yyyy
john	doe	doe1220	
Address	City	State	Country
1344	N WILLOW AVE	Alabama	USA
Zipcode	Phone	Purchase Order Number	
74014	(249) 220-9999		

Below the form fields, there are sections for "Type of Payment (For Orders without PONumber)" and "Shipping".

**Type of Payment (For Orders without PONumber)**

- Visa
- Discover
- Master Card
- Check
- PONumber already selected

**Shipping**

- Fedex
- OverNight Fedex
- Regular Mail

At the bottom of the form, there are two buttons: "Continue" and "clear form".

Figure (3.4)  
Order Form

### **3.1.3 Save Order Details Corresponding to an Order**

The order detail form receives information such as name of sample, type of sample, name of test, volume of DNA, concentration of DNA, and type of primer for each sample. A form can accept up to four samples at a time. The details are

submitted using the “add more samples” button if additional forms are needed. If a customer has entered all his samples he uses the “ save and exit” button. Both the buttons trigger the same CGI program. Figure (3.5) displays the form for accepting samples for each order. Figure (3.6) displays the billing statement for the order.

### **3.1.3.1 orderdetail.cgi**

This executable file is designed to read data from the order detail form. It consists of code that repeatedly scans consecutive rows of the order detail form. During each loop the SQL procedure **savesampledetail** is executed to write the sample detail information to the database table **sampledetail**. Each record is identified by a unique identity *sampledetail.sampledetailid* which is a numerical identity generated automatically by the Sybase database. The CGI program prepares a billing statement when the order detail form is completed. In order to create a virtual bill all the records corresponding to the order are read from the **sampledetail** and **ordertable** database tables. It then computes the total sum of unit price, sales tax, and freight charge to generate a detailed bill of expenses. If “add more samples” button triggers the CGI program then it generates additional forms for the customer and repeats the same procedure until he submits the form using a “save and exit” button.

### 3.1.3.1 savesampledetail( @orderid char(15), @samplename varchar(20),

@typeofsample varchar(20), @volofdna varchar(10), @conc varchar(10), @primer varchar(20), @testname varchar(20), typeoftest varchar(20) )

This SQL procedure uses @testname to identify the price charged to the customer for the test. The datatypes of @volofdna and @conc are then converted into a numeric format and the sample detail information is inserted as a new record into the database table **sampledetail**.

The screenshot shows a Netscape browser window titled "Orderdetail form in process". The browser's menu bar includes File, Edit, View, Go, Communicator, and Help. The toolbar contains icons for Back, Forward, Reload, Home, Search, Netscape, Print, Security, and Shop. The location bar shows the URL "http://localhost/cgi-bin/process\_order.cgi". Below the browser window, the form content includes the following text and table:

Fill out the following details for each of the samples

For convenience of processing samples in batches customers are expected to have a minimum of 4 sample ,and any more samples should be multiples of 4

This orderid should be used for all future inquiry

00E1191

samplename	sample type	DNA vol	Conc	Primer	Reaction	Test Types
00E112	Plasmid	10	10	M13Rev	ST	Reverse
00E113	Bac	10	20	T3	TS2	Reverse
00E114	Bac	10	30	T3	ST	Forward
00E115	Plasmid	10	40	M13Rev	TS2	Forward

Buttons: Save and Exit, Add More Samples

Figure (3.5)  
Order Detail Form

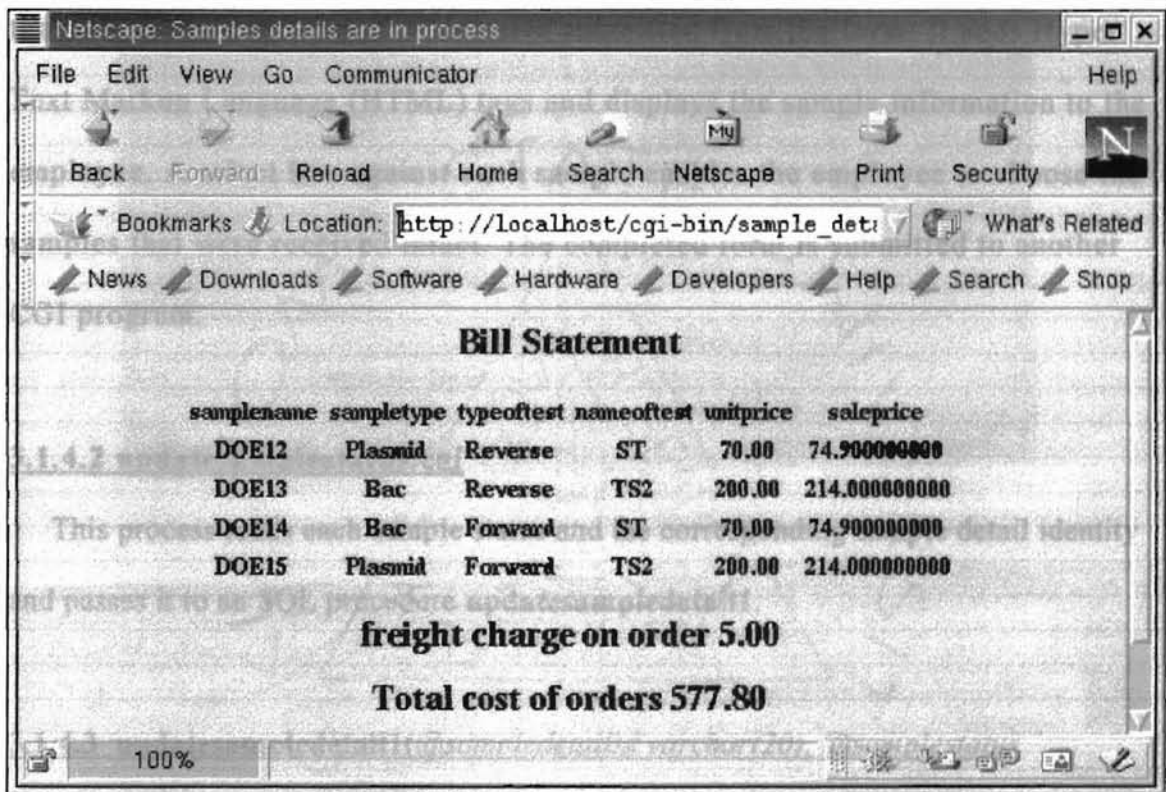


Figure (3.6)  
Billing Statement  
Figure (3.5)  
Order Detail Form

### 3.1.4 Update Sample Status when Samples are Received by Mail

When samples are received by mail at the corporate office they are checked for damages and the status of column *sampledetail.samplereceived* is set to “yes” when samples are intact. This is necessary because samples are moved to the worktable only when the samples arrive without any damage. The employee submits *ordertable.orderid*, and *customers.customerid* to track the sample details posted electronically. A CGI program reads the data from the database table and displays it as a form. Figure (3.7) displays the form for retrieving records from table. Figure (3.8) displays the form used to update sample received status.

#### **3.1.4.1 samplestatusupdate.cgi**

This CGI program retrieves the records from the database table **sampledetail** that match the *orderid* and *customerid* submitted from the form. It adds Hyper Text Markup Language (HTML) tags and displays the sample information to the employee. A select box against each sample enables the employee to choose the samples that were received intact. The completed form is submitted to another CGI program.

#### **3.1.4.2 updatesamplestatus.cgi**

This process reads each sample status and the corresponding sample detail identity and passes it to an SQL procedure **updatesampledetail1**.

#### **3.1.4.3 updatesampledetail1(@sampledetailid varchar(20), @samplestatus varchar(5))**

This procedure receives its parameters from the CGI program **updatesamplestatus.cgi**. The parameter *@samplestatus* has a value "yes" when the sample unit is received at the corporate office without damage. The field value *sampledetail.samplereceived* on database table **sampledetail** corresponding to *@sampledetailid* is updated with the new value of *@processstatus*.



Figure (3.7)  
Select Records for Posting Arrival

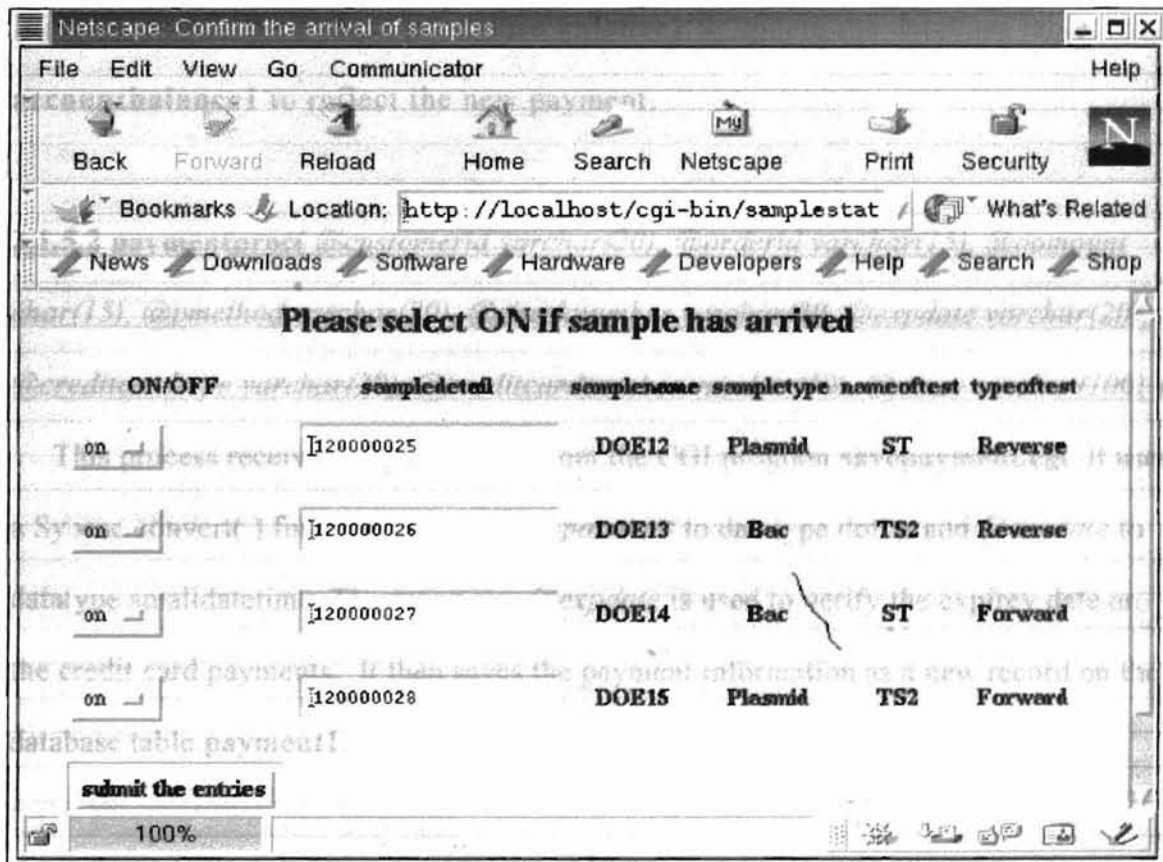


Figure (3.8)  
Update Arrival

### **3.1.5 Post Payment Information to Database.**

When checks or credit card information from customers are received at the central office, all the data pertaining to the payment are recorded in the database table **payment1**. Using a form an employee submits the customerid, orderid, type of payment, credit card number, date of payment, payment amount, ...etc to a CGI program. Figure (3.9) displays the form for recording payments received.

#### **3.1.5.1 savepayments.cgi**

This CGI program reads the field values from the payments form. It executes an SQL procedure **paymentproc** to save the data to the database table **payment1**. It also executes the Sybase procedure **balance1** to update the database table **accountbalance1** to reflect the new payment.

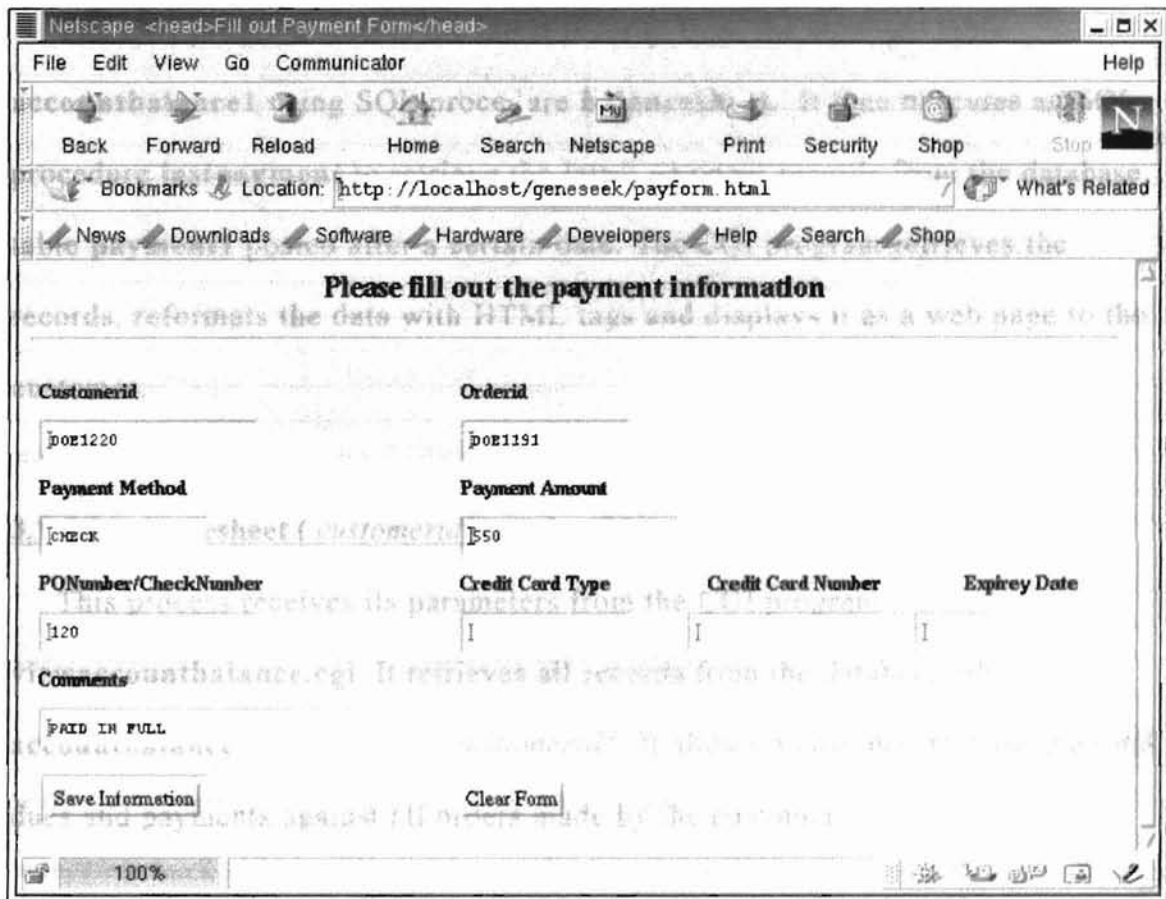
**3.1.5.2 paymentproc( @customerid varchar(20), @orderid varchar(15), @pamount char(15), @pmethod varchar(20), @checknumber varchar(30), @expdate varchar(20), @creditcardtype varchar(20), @creditcardnumber varchar(30), @notes varchar(100) )**

This process receives its parameters from the CGI program **savepayment.cgi**. It uses a Sybase convert( ) function to convert *@pamount* to datatype dollar and *@expdate* to datatype smalldatetime. The parameter *@expdate* is used to verify the expiry date on the credit card payments. It then saves the payment information as a new record on the database table **payment1**.

**3.1.5.3 balance1( @orderid1 varchar(15) )**

This process receives its parameters from the CGI program **savepayment.cgi**. It computes the current dues for a certain *@orderid1* by

calculating the total cost of all the tests, sales tax and freight charge from the database table **sampledetail**. The sum total of all the payments are computed from the database table **payment1** for the same **@orderid1**. The dues and payments for the **@orderid1** and **@customerid1** are updated to the database table **accountbalance1**.



The screenshot shows a Netscape browser window with the title "Fill out Payment Form". The address bar displays "http://localhost/geneseek/payform.html". The form contains the following fields and controls:

<b>Customerid</b>	<b>Orderid</b>		
<input type="text" value="DOE1220"/>	<input type="text" value="DOE1191"/>		
<b>Payment Method</b>	<b>Payment Amount</b>		
<input type="text" value="CHECK"/>	<input type="text" value="\$50"/>		
<b>PONumber/CheckNumber</b>	<b>Credit Card Type</b>	<b>Credit Card Number</b>	<b>Expirey Date</b>
<input type="text" value="120"/>	<input type="text" value="I"/>	<input type="text" value="I"/>	<input type="text" value="I"/>
<b>Comments</b>			
<input type="text" value="PAID IN FULL"/>			
<input type="button" value="Save Information"/>		<input type="button" value="Clear Form"/>	

Figure (3.9)  
Payment Form

### 3.1.6 Review Account History

A customer or partner can use this application to request a document showing the latest account activity and review his latest account balance. A virtual



document displays all payments posted after the date specified and account balance information corresponding to each order. A customer submits his identity and a specific date to the CGI program to retrieve his records. Figure (3.10) displays the latest account transactions and account balance.

#### **3.1.6.1 viewaccountbalance.cgi**

The CGI program selects the latest records from the database table **accountbalance1** using SQL procedure **balancesheet**. It then executes an SQL procedure **lastpayment** to retrieve the latest payment records from the database table **payment1** posted after a certain date. The CGI program retrieves the records, reformats the data with HTML tags and displays it as a web page to the customer.

#### **3.1.6.2 balancesheet ( customerid1 varchar(20) )**

This process receives its parameters from the CGI program **viewaccountbalance.cgi**. It retrieves all records from the database table **accountbalance1** that match *@customerid1*. It allows a customer to view the total dues and payments against all orders made by the customer.

#### **3.1.6.3 lastpayment( @customerid1 varchar(15), @date1 varchar(15) )**

This process receives its parameters from the CGI program **viewaccountbalance.cgi**. It retrieves all records from the database table **accountbalance1** that match the parameter *@customerid1*. This allows the customer to see all the payment records from database table **payment1** made by *@customerid1* posted after *@date1*.

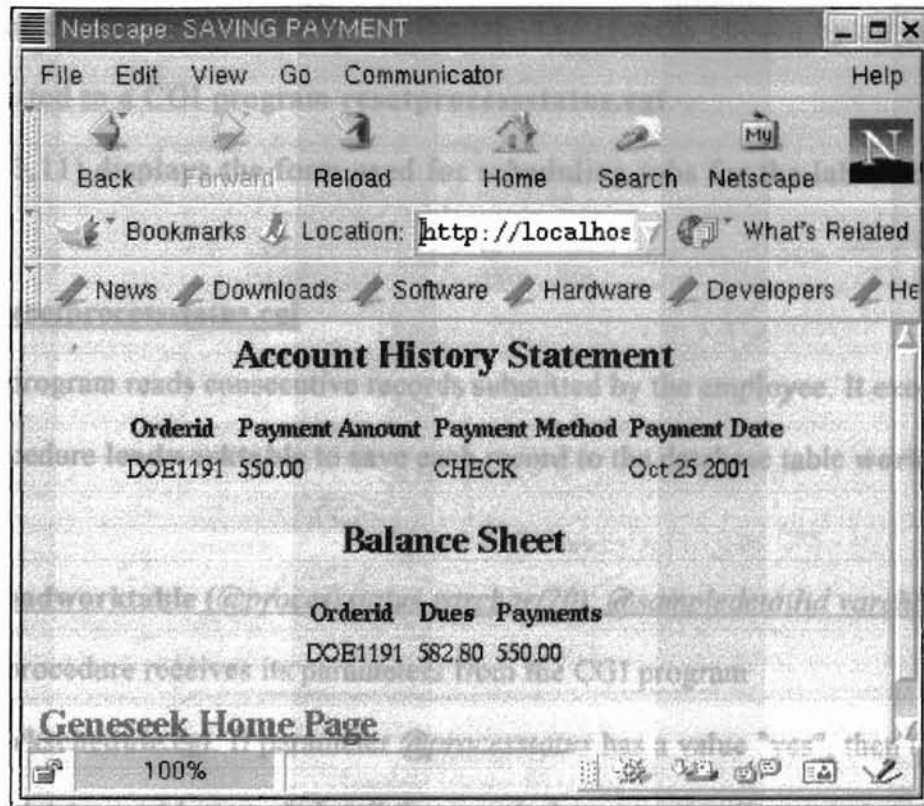


Figure (3.10)  
Account Information

### **3.1.7 Schedule a Batch of Samples for Testing**

In response to a request from the laboratory, all records from a **sampledetail** table that qualify for immediate testing are selected. A CGI program **selectworkschedule.cgi** executes the SQL commands required to retrieve records from the database table **sampledetail** that have arrived by mail and are waiting to be tested.

#### **3.1.7.1 selectworkschedule.cgi**

This program executes an SQL script to select records that qualify for testing from the database table **sampledetail**. It reads records, reformats the columns,

adds HTML tags and displays it as a virtual form. A select option allows the employee to choose the samples for the lab. The records chosen by an employee is submitted to a CGI program **resetprocessstatus.cgi**.

Figure (3.11) displays the form used for scheduling jobs for the laboratory.

### **3.1.7.2 resetprocessstatus.cgi**

This program reads consecutive records submitted by the employee. It executes the SQL procedure **loadworktable** to save each record to the database table **worktable**.

### **3.1.7.3 loadworktable (@processstatus varchar(20), @sampledetailid varchar(20))**

This procedure receives its parameters from the CGI program **selectworkschedule.cgi**. If parameter *@processstatus* has a value "yes", then the record from the database table **sampledetail** that match the parameter *@sampledetailid* are copied to the database table **worktable**. It then resets the columns *worktable.processstatus*, *sampledetail.processstatus* to "IP" and *worktable.processstartdate* to current date to indicate that testing has been initiated.

### **3.1.8 Update Test Results to Database**

An employee uses this process to save file location of results and comments to the database. Test results are stored as a file. An employee uses a form to initiate a CGI program that selects all records on the worktable that are awaiting results. Figure (3.12) displays the form used for saving results and file locations to database.

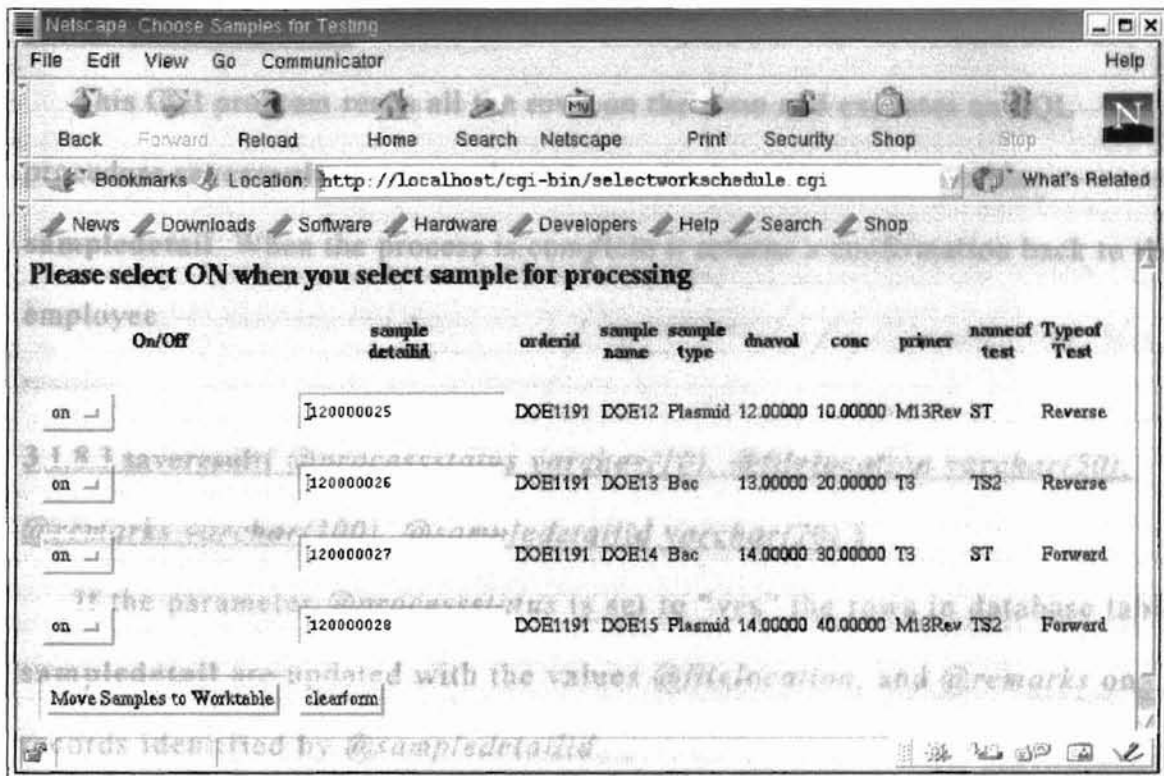


Figure (3.11)  
Schedule Samples for testing

### 3.1.8.1 updateresultstoworktable.cgi

The CGI program executes an SQL query that selects records on the database table **worktable** that are awaiting results. The records retrieved from the ISQL interface are reformatted with suitable HTML tags and displayed as a virtual document to the employee. Text fields are added for entering location of files, and to enter specific remarks about the test. Using a select option an employee chooses "yes" for all documents whose results are complete. He also enters the new file location and comments corresponding to each test. The completed form is submitted to another CGI program.

### 3.1.8.2 saveresults.cgi

This CGI program reads all the rows on the form and executes an SQL procedure **saveresults** to save selected rows back to the database table **sampledetail**. When the process is complete it returns a confirmation back to the employee.

### 3.1.8.3 saveresult( @processstatus varchar(10), @filelocation varchar(50), @remarks varchar(100), @sampledetailid varchar(20) )

If the parameter *@processstatus* is set to “yes” the rows in database table **sampledetail** are updated with the values *@filelocation*, and *@remarks* on records identified by *@sampledetailid*.

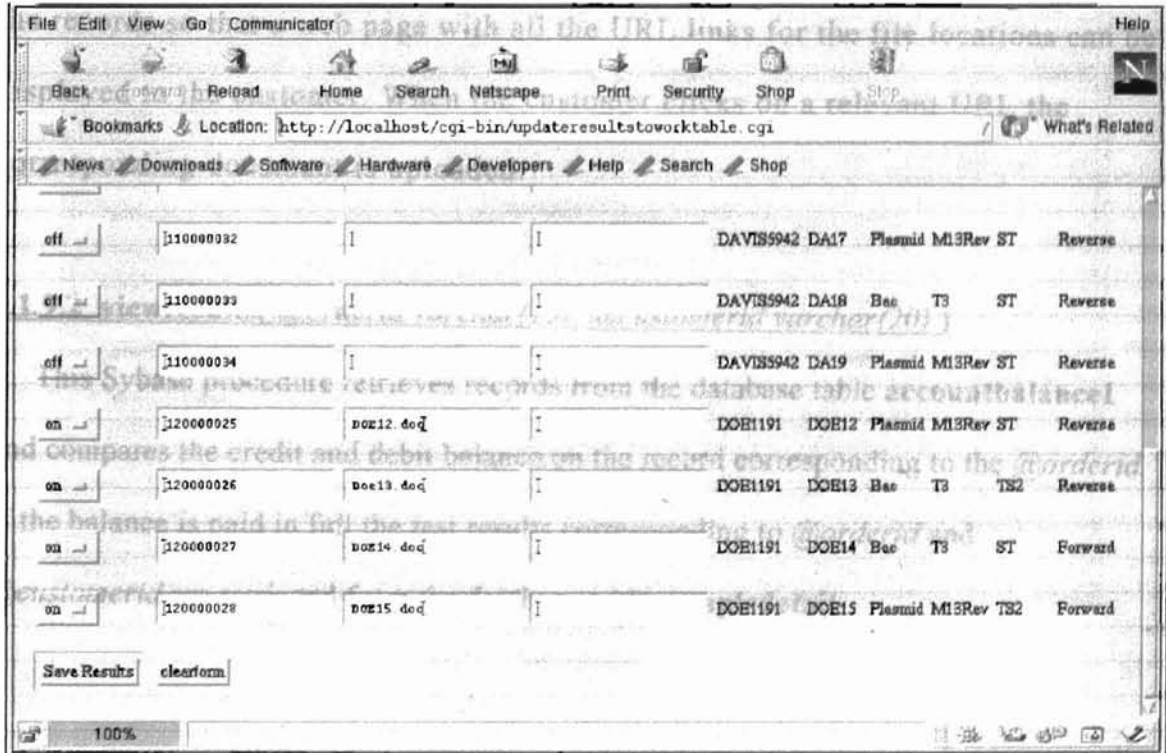


Figure (3.12)  
Save Results to Database

### **3.1.9 View Summary of Results**

Customers can use this process to access test results, and observations electronically. He submits his *orderid* and *customerid* to a CGI program. Figure (3.13) displays a form for accepting information for tracking results. Figure (3.14) displays the outcome of tests and links to the files containing test results.

#### **3.1.9.1 viewresults.cgi**

This CGI program executes an SQL procedure **viewresults** to verify if payments have been made in full for the order. If there is no negative balance, all the files locations containing test results and remarks about the test are retrieved from the database table **sampledetail**. Appropriate HTML tags are attached to the records so that a web page with all the URL links for the file locations can be displayed to the customer. When the customer clicks on a relevant URL the corresponding document is uploaded.

#### **3.1.9.2 viewresults( @orderid varchar(15), @customerid varchar(20) )**

This Sybase procedure retrieves records from the database table **accountbalance1** and compares the credit and debit balance on the record corresponding to the *@orderid*. If the balance is paid in full the test results corresponding to *@orderid* and *@customerid* are retrieved from the database table **sampledetail**.

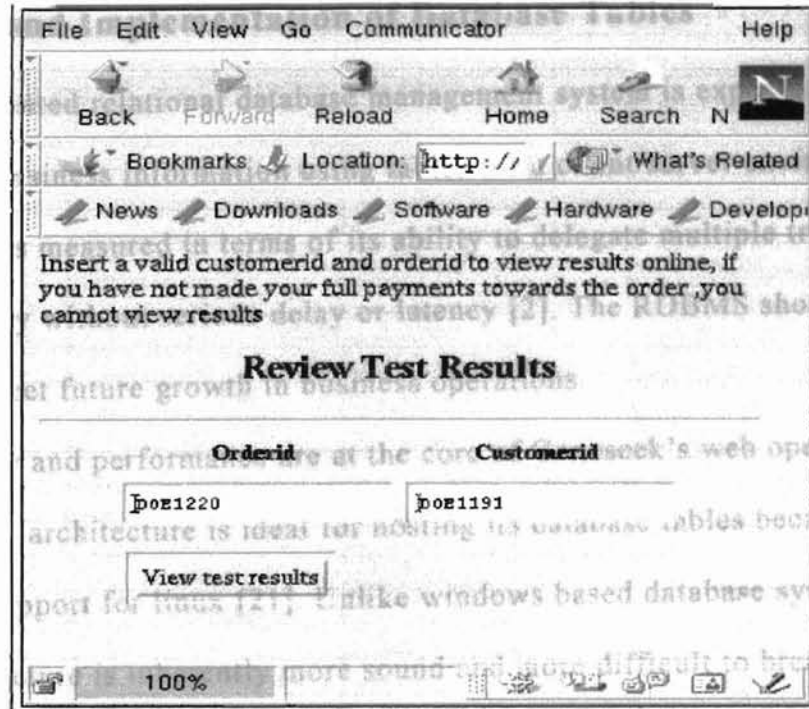


Figure (3.13)  
Retrieve Records of  
Results

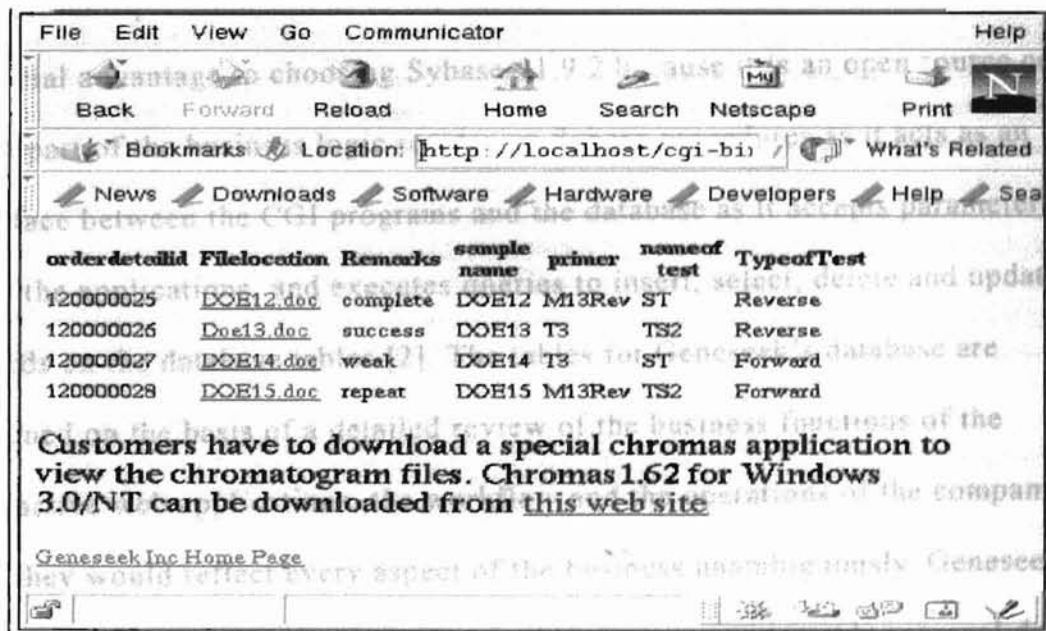


Figure (3.14)  
Test Results

### **3.2 Design and Implementation of Database Tables**

So illustrate

A web oriented relational database management system is expected to store and manage business information using tables for a client/server environment. Its performance is measured in terms of its ability to delegate multiple transactions simultaneously without serious delay or latency [2]. The RDBMS should also be scalable to meet future growth in business operations.

Since security and performance are at the core of Geneseek's web operations, Sybase 11.9.2 architecture is ideal for hosting its database tables because of its underlying support for linux [21]. Unlike windows based database systems, linux based architecture is inherently more sound and more difficult to break into [20]. In addition it also has a virtual memory architecture that enhances its ability to support multiple connections to the database server. Moreover there is added financial advantage to choosing Sybase-11.9.2 because it is an open source code.

A part of the business logic resides on Sybase procedures as it acts as an interface between the CGI programs and the database as it accepts parameters from the applications, and executes queries to insert, select, delete and update records on the database tables [2]. The tables for Geneseek's database are designed on the basis of a detailed review of the business functions of the interactive web applications, the workflow and the operations of the company so that they would reflect every aspect of the business unambiguously. Geneseek's database consists of a group of related tables to store customer's personal data, account information, balance sheet, inventory, and order logs and work status. A one-to-many relationship is maintained between a primary key of the parent table and the foreign key of child table to maintain referential integrity so that SQL



statements can join records from parent and child tables. Figure (3.15) illustrates the relationship established between the tables.

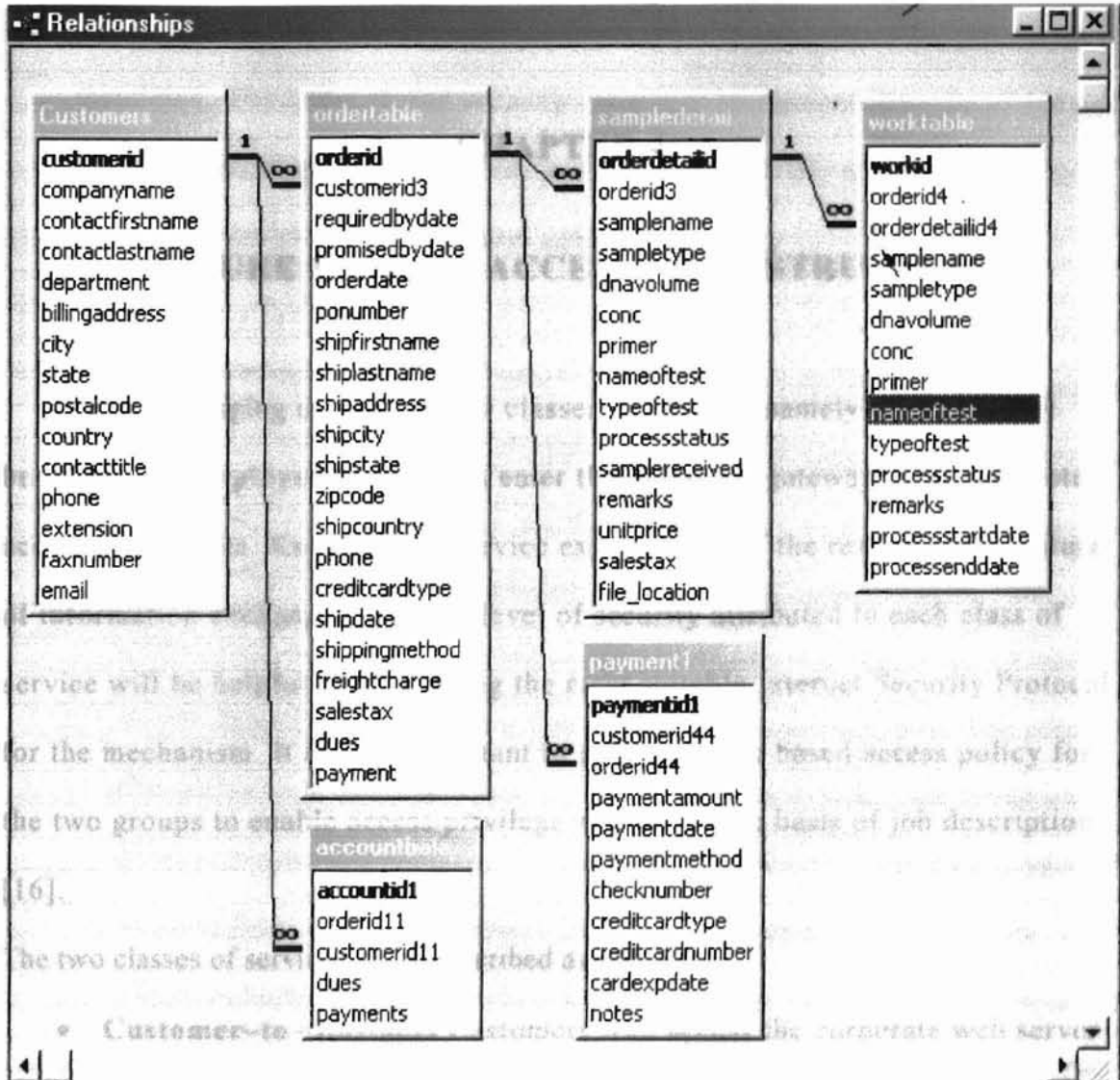


Figure (3:15)  
Relationship Between Database Tables

## CHAPTER 4

### SECURE REMOTE ACCESS INFRASTRUCTURE

Traffic belonging to two distinct classes of service, namely client-to-business, and employee-to-business enter the corporate gateway using a remote access mechanism. Knowing the service expectations of the remote users, nature of information exchanged, and the level of security attributed to each class of service will be helpful in identifying the most suitable Internet Security Protocol for the mechanism. It is also important to enforce a role based access policy for the two groups to enable access privilege strictly on the basis of job description [16].

The two classes of service can be described as follows:

- **Customer-to –Business** Customers who access the corporate web server require limited access to the customer database residing behind the firewall for occasional queries or updates. Each remote workstation may be dependant on an analog modem for Internet connectivity. The traffic is intermittent and not data intensive. File exchanges are few and far between. Whenever such connections are enabled they should support a minimum level of security, and file exchanges should be done in total confidentiality. Most hacker attacks are launched from unsecured remote hosts. Once the web server is compromised hackers can take over the

internal network. Although a customer cannot be expected to install expensive devices or software for authentication some basic measures can be mandated. The network and security protocols implemented at the client and corporate servers should have adequate interoperability and scalability to support future growth in business operations.

- **Employee-to-Business** An employee that goes outside the network needs free access to the servers and applications and he expects the same level of service from outside. The access privileges are unlimited and such sessions should be strictly monitored. These employees require read, write access to all database tables. The most suitable option would be to isolate this service on private lines, but that is too expensive. The other alternative is to provide strong password authentication, encryption and diligent monitoring features. This service is limited to employees who are working from permanent remote locations such as home or a remote office. A higher degree of financial commitment can be expected from the employee in terms of high-speed connectivity, local implementations for network level encryption, and customer premise equipment. Traffic is data intensive and requires higher computation power from both corporate servers and remote hosts. This form of network access can be purchased from Internet service providers or it can be installed at the customer premises.

## **4.1 Internet Security Protocol Alternatives**

Web enabled applications can be utilized fully only if there is an effective yet safe mechanism for communication from remote sites. The traditional private line approach is no longer a viable alternative considering the number of customer sessions supported simultaneously, instead SSL and IPSec protocols offer a more affordable solution by utilizing the publicly available infrastructure to build virtual and secure communication lines from remote locations to the central office [8]. SSL and IPSec use similar means to ensure that attributes such as authentication, encryption, authorization, and message integrity of business transactions are preserved. Although they have the same safety features they address entirely different business environments. The performance of SSL and IPSec protocol implementations on corporate sites are compared in terms of ability to support heterogeneous client environments, ability to support multiple clients, ability to maintain predictable performance under high traffic volume, and ability to scale to meet increased traffic and new applications. A remote access protocol is recommended based on how well it meets the company specifications.

### **4.1.1 Implications of Using SSL**

SSL applies security below the application layer and above the TCP layer and so all applications that run over TCP such as FTP, TELNET, HTTP and EMAIL can share the same SSL security infrastructure [3]. This is an ideal proposition for customers and employees who access multiple services from time to time from their homes. The client implementation of SSL is preloaded in almost all

windows based web browsers. SSL protocols offer strong encryption and authentication mechanism for its price and are ideal when business transactions are few. SSL establishes secure context sensitive connections between a client desktop and a corporate server [5]. This ensures end-to-end security during a secure business transaction. However SSL is very slow when it has to support high-speed bulk transactions because packet overheads are very high when security is applied above the transport layer [3]. SSL protocols are independent of the IP address and so a client can access the corporate server from locations such as Internet café's and laptop computers. Since the SSL protocol does not require an IP address to identify a client it can support a network address translator at the corporate site [13]. Most of the security intelligence will reside on the client's desktop and the corporate web server.

This is a very primitive networking philosophy and it is plagued by packet overheads, and excessive delay because security encapsulation is applied above the transport layer [3]. Moreover SSL protocols are limited to the transport layer and the degree of flexibility enjoyed by network level security protocols are lacking in SSL which explains the reason why SSL cannot be implemented in complicated network settings where intranet connections are needed between LAN's [5]. SSL protocol is also prone to network level security violations that remain undetected if security is applied above the transport layer [10, 17]. The overhead of key negotiation is considerably large and consequently results in delays. Although reusing cached handshakes by consecutive sessions reduces latency, the delays due to packet overheads far outweigh the benefits of caching. SSL has severe limitations when it is applied to a complex networking

environment that supports thousands of concurrent connections and in some cases a network level LAN-to-LAN connection [17].

#### **4.1.2 Implications of Using IPSec**

IPSec, on the other hand, applies a more sophisticated form of security by encrypting and encapsulating data packets at the network layer on the TCP/IP stack. Large networks that support multiple services can use the same network level security infrastructure on all forms of traffic irrespective of the applications where the traffic originates, and the types of transport protocols used by the traffic. In addition IPSec based VPN solutions are offering the latest techniques and protocols to effortlessly manage thousands of users scattered worldwide. New versions of VPN also support NAT, dynamic key infrastructure for exchanging public keys, automatic client configuration, and firewall policies at the network level to enhance the service features available on the same box. Using a database of security associations or SA's, IPSec can allocate different security features to individual users based on the type of header chosen to protect the packets. Figure (2.3) illustrates the choices available for IP traffic. IPSec protocols have the flexibility of operating in two modes namely transport mode and tunnel mode. If IPSec protocols are applied to the transport layer it is said to be in transport mode, if it is applied at the network layer it is said to be in tunnel mode. In transport mode IPSec offers security for host-to-host communication. Tunnel mode on the other hand operates even on intermediate nodes located between the ultimate destinations. When IPSec protocols operate in tunnel mode it can offer services for multiple forms of communication including host-to-host, gateway-to-gateway, and host-to-gateway [9, 5]. These

unique encapsulation features allow IPsec to support different types of network connections in a complex network setting. Since security encapsulation is applied at the network layer, at the lower end of the TCP/IP stack, packet overheads are considerably reduced [5]. Consecutive sessions set up between the same end points can avoid delays associated with CPU intensive handshake protocols at the beginning of each session by reusing session handshake information from previous sessions [10].

The main concern with IPsec solution is that each vendor implements his own interpretation of the IPsec standards causing incompatibility between different versions of the same security standard [8]. These issues are typical of any evolving technology because the standards are continuously changing [8]. Once the IPsec standards are established throughout the industry, the lack of interoperability can be minimized. Enterprises can run into difficulties if hardware at the client site does not support the VPN software correctly. These configuration issues can be solved to an extent by automatically configuring the client environment and implementing the necessary IPsec client software as part of the connection process [6]. Internet Security Association and Key Management Protocol (ISAKMP) protocols located on the corporate server perform the automated client configuration. This is a very convenient feature when a large number of customer accounts must be activated at short notice.

Another serious aspect of the client implementation is the difficulty in placing security restrictions on the desktops used for remote access [15]. If employees access the Internet through DSL cable modems they are part of a local service provider loop that is always connected to the public network. Hackers can guess the range of IP addresses available and run scans to detect targets that are connected. Once they hit a



target they use it to launch attacks on the corporate server. The lack of security at the client desktop can be addressed by enforcing some simple measures such as personal firewalls to detect viruses and warn of suspicious hacker activity, restrictions on downloading vulnerable soft ware, and strict policies for storing proprietary corporate information [15].

IPSec standards can be deployed effectively in a virtual private network setting with some foresight and planning. Based on the of the evaluation conducted during the course of this work, IPSec has emerged as an ideal choice for the VPN infrastructure that serves Geneseek, because of its ability to manage a large number of clients effectively, and to maintain separate security policies for its customers and employees.

## **4.2 Authentication and Authorization of Users**

Determining an ideal security protocol is only a part of the solution. There are other steps necessary to authenticate and authorize users to access the web services. Current IPSec implementations at the corporate center offer infrastructure to authenticate using public keys or a pre-shared secret key. In the case of a public key it needs Public Key Infrastructure (PKI) to issue certificates containing public keys for first time clients, and to verify the validity of public keys submitted on public certificates when the client requests a connection [7]. When pre-shared keys are used instead of public keys, either RADIUS or Novell Directory Service is necessary to manage and store pre-shared keys belonging to customers.

PKI is an automated mechanism for managing large-scale operations for a worldwide audience because certificates can be issued, and sent to the client electronically in virtually no time. On the other hand, a password or pre-shared key must be generated and stored on the RADIUS server manually. It is then sent to the client on a floppy disk. The user must load the client implementation before accessing the corporate server. The server verifies the password, and then allows the customer to access the web service. This process is error prone, and difficult to manage when thousands of users need access to the corporate office.

A PKI model is more appropriate for authenticating Geneseek's customers and clients, because its clientele is expected to grow exponentially. In keeping with the level of secrecy associated with Geneseek's transaction it is safer to supervise the PKI servers within the company.

Maintaining separate passwords to access each server on the network is ineffective in a complex network environment that hosts multiple services [16]. A more suitable alternative is to use a single centralized authority at the network level to differentiate access privileges based on the class of service associated with a specific user name or traffic flow. Once the clients are identified positively they are allowed to exercise their access rights. When these clients retrieve records from the corporate server or send data to the server, packets are secured based on the attributes defined by the SA policy [5]. Policies for clients are at a central database. A mechanism such as a Light Weight Directory Access Protocol (LDAP) is used to distribute policy from the central database to the remote clients and the central VPN server when connections are requested [5,16].

### **4.3 Message Encryption and Authentication**

An IPsec based VPN solution is only as strong as its encryption algorithm. Such algorithms encrypt data to ensure end-to-end confidentiality. Hence the symmetric algorithms for encryption and message authentication cannot be compromised. During an initial handshake, peers negotiate the algorithms for encryption, verify the authenticity of user identity, and exchange the symmetric keys for the algorithms. Strong algorithms are selected for message encryption and message authentication to make it increasingly difficult for hackers to guess the key and decipher the actual text [5]. Hence the data receives a higher degree of protection against being exposed.

Message authentication is a measure taken to detect if encrypted text has been tampered during transit. Running the actual message through a hash algorithm creates a Message Authentication Code (MAC). MAC is then sent along with the encrypted message to the receiving end. The recipient decrypts the actual message and then runs the text through a hash algorithm. The MAC's created are identical if the message remains intact throughout transit. Most IPsec implementations support a keyed hash (HMAC) because they generate hashes in less time and are cryptographically stronger [5].

### **4.4 Network Address Translation**

Web based business units offer a number of services to its customers. When more servers interact with the public, there is a greater need for unique IP addresses than before. Enterprises cannot limit Internet connections to servers, personnel computers, and other devices because of the scarcity of public IP

HTTP, and FTP [19]. Functions also include content filtration of traffic for certain URL destinations, detecting viruses embedded in encrypted messages, and maintaining detailed reports on client activity such as length of session, data transferred, etc [20]. Firewalls and IPSec protocol implementations are sometimes located on the same box because the two entities complement each other in terms of security. It has been found to more beneficial to apply a firewall policy behind the IPSec infrastructure because the firewall cannot filter the message content unless they are decrypted [6].

## CHAPTER 5

### SUMMARY AND CONCLUSION

This thesis has the objective of solving a unique business problem for a biotechnology company named Geneseek Inc. The company aspires to be a pioneer in the biotech world, by launching the first virtual laboratory on the World Wide Web. This project is a very big undertaking considering the scope for customer-to-business, employee-to-business, and business-to-business operations that could be launched over the web.

This work focuses on understanding the flow of operations that take place at the corporate office to identify efficient solutions for automating the customer centered operations that would otherwise be time consuming and error prone when handled by customer representatives. These solutions are aimed at transferring some control to the customer to give the customer the convenience of placing orders, tracking the status of the orders and payments, downloading results, etc from his office. The solution also offers a variety of services to employees who wish to manage the back office system remotely.

These applications currently are running in a Linux based environment loaded on a Pentium system. An Apache Server loaded on the same device hosts the application including the web pages and the executable CGI programs. The CGI programs for these applications were developed in a Linux environment using C. A Sybase-11.9.2 RDBMS

system was configured and implemented in the Linux environment to support the back office system.

The company elected to adopt an Internet based remote access solution in light of the prospects of secure, affordable and effective service comparable to that of a private line. During the course of this project the various alternatives for connecting the clients and employees to the corporate center were explored and a set of techniques and guidelines that would be most appropriate for guarding the company's network perimeter were identified and recommended.

The actual business applications launched by the corporate business infrastructure will be an enhanced adaptation of the working model created by this project. When the applications are moved to the corporate site, services can be accessed through a web browser from the client location. Additional services can be easily incorporated to add services like inventory management and business-to-business transactions as the technology evolves to simplify directory-to- directory transfer of data in different semantics. The facts about VPN protocols clearly suggest that it cannot be limited to connecting clients to the corporate office. VPN protocols are improving to extend intelligence and security beyond the access gateways to include intermediate routers and switches. This creates immense possibilities for expanding the corporate network so that partners and branch offices can freely access information resources and services located at the corporate office.

## REFERENCES

1. Gundavaram, S. CGI Programming on the World Wide Web. O'Reilly&Associates, Inc., Sebastopol, CA. 1996.
2. Anderson, G. W. Client/Server Database Design with Sybase: A high performance and fine-tuning guide. McGraw-Hill., New York, NY. 1997.
3. Apostolopoulos, G., Peris, V., Pradhan, P., Salva, D. "Securing Electronic Commerce: Reducing the SSL overhead." IEEE Network, Vol. 14 (4), 2000.
4. Gilmore, C., Korman, D., Aviel, D. R. "Secure Remote Access to an Internal Web Server." IEEE Network, Vol. 14 (4), 2000.
5. Doraswamy, N., Harkins, D. IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks. Prentice Hall PTR., Upper Saddle River, NJ. 1999.
6. Browne, B., Lewis, C., Hamilton, R., Weaver, W. "Best Practices for VPN Implementation." BCR Communications Review, Vol. 31(3), 2001.
7. Perlman, R. "An Overview of PKI Trust Models." IEEE Network, Vol. (), 1999.
8. Mier, E. R., Smithers, J. R., Maksymov, J. M., "Making VPN's Work." Business Communications Review, Vol 29(11), 1999.
9. Borella, S. M. "Methods and Protocols for Secure Key Negotiation using IKE." IEEE Network, Vol. 14(4), 2000.
10. Perlman, R., Kaufman, C. "Key Exchange in IPSec: Analysis of IKE." IEEE Internet Computing, Vol. 4(6), 2000.
11. Mancill, T. "Linux in the Corporate Network?." Business Communications Review, Vol. 30(1), 2001.
12. Liebmann, L. "To Have And Have NAT: Managing Through the Firewall." Business Communications Review, Vol. 29(7), 1999.
13. Shieh, S., Ho, F., Huang, Y., Luo, J. "Network Address Translators: Effects on Security Protocols and Applications in the TCP/IP Stack." IEEE Internet Computing, Vol. 4(6), 2000.

14. Younglove, R. "Virtual Private Network: Secure Access for E-Business." IEEE Internet Computing, Vol. 4(4), 2000.
15. King, M. C. "Remote Access VPN's: Selection and Deployment Issues." Business Communications Review, Vol. 30(6), 2000.
16. Gutzmann, K. "Access control and Session Management in HTTP Environment." IEEE Internet Computing, Vol. 5(1), 2001.
17. Farrow, R. "Secure Socket Layer is not a Magic Bullet." Network Magazine, Vol. 16(1), 2001.
18. Tiwani, A., Balasubramaniam, R. "Integrating Knowledge on the Web." IEEE Internet Computing, Vol. 5(3), 2001.
19. King, M. C. "Will the Enterprise Perimeter Survive." Business Communications Review's Access, August 2000.
20. Blancharski, D. "Create Order With a Strong Security Policy." Network Magazine, Vol. 15(7), 2000.



VITA

Sarah Cheriyan

Candidate for the Degree of

Master of Science

Thesis: A WEB-BASED INFORMATION TECHNOLOGY SOLUTION FOR  
A SMALL BUSINESS

Major Field: Computer Science

Biographical:

Personal Data: Born in Kerala, India, the daughter of Achamma and  
Kattil Punnose Cheriyan.

Education: Graduated from University of Kerala, Kerala, India, with a  
Bachelor of Engineering degree in Civil Engineering in December,  
1995. Completed the requirements for the Master of Science degree in  
Computer Science at Oklahoma State University in December, 2001.

Experience: Employed by the Directorate of Technical Education as an  
Instructor for Computer Aided Design (CAD) at the College of  
Engineering, Kerala, India in Spring 1996.

Professional Memberships: IEEE Student Chapter.

