KUMARAGURU
COLLEGE OF TECHNOLOGY
Department of Computer Science and Engineering

ISO 9001:2000 Certified

INTRANET AND INTERNET MAIL SERVER

PROJECT WORK DONE AT
ADVANCE ECOM SOLUTIONS P - 1243

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF
M.SC APPLIED SCIENCE (SOFTWARE ENGINEERING)
OF BHARATHIAR UNIVERSITY, COIMBATORE.

Submitted by

B.RAJKUMAR
Reg.No-0137S0047

Guided by

INTERNAL GUIDE

Mr.R.Rajasehar, M.C.A.,
Lecturer,
Dept. of Computer Science & Engineering,
Kumaraguru College of Technology,
Coimbatore.

EXTERNAL GUIDE

Mr. B.Ramesh, M.C.A.,
CEO,
Advance Ecom Solutions,
Coimbatore.

CERTIFICATE

KUMARAGURU
COLLEGE OF TECHNOLOGY
Department of Computer Science and Engineering

ISO 9001:2000 Certified

CERTIFICATE

PROJECT REPORT SEPTEMBER 2004

Certified that this is a bonafide report of

The project work done by

B . RAJKUMAR

Reg.No-0137S0047

Prof.Dr.S.Thangasamy, B.E, (Hons), Ph.D.,	**Mr. R .Rajasehar, M.C.A.,**
Professor and Head	Project Guide
Computer Science & Engineering	Computer Science & Engineering

Place: Coimbatore
Date:

Submitted for University examination held on

30.09.04

Internal Examiner

External Examiner

COMPANY CERTIFICATE

Date: 20-09-2004

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. B. RAJKUMAR** (Reg.No. **0137S0047**) of 4th year **M.Sc.(Software Engineering)**, Kumaraguru College of Technology, Coimbatore has successfully completed the project work titled "**INTRANET AND INTERNET MAIL SERVER"**. He has done the project from June 2004 to September 2004. During the period of the project, he was found to be hard working and sincere in his assignments.

B.Ramesh
(CEO)

DECLARATION

DECLARATION

I here by declare that the project entitled **"INTRANET AND INTERNET MAIL SERVER",** submitted to **Kumaraguru College of Technology**, coimbatore Affiliated to Bharathiar university as the project work of **Master of Science in Applied Science Software Engineering,** is a record of original work done by me under the supervision and guidance of **Mr.B.Ramesh ,M.C.A.,** Advanced Ecom Solutions, Coimbatore and **Mr.R.Rajasehar, M.C.A.,** Lecturer of Kumaraguru College of Technology, Coimbatore and the project work has not found the basis for the award of any Degree/Diploma/Associate ship/Fellowship or similar title to any candidate of any University.

Place: COIMBATORE
Date: 27.09.04

Signature of the student

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I am immensely grateful to **Dr.K.K.Padmanaban B.Sc (Engg), M.Tech., Ph.D.,** Principal, Kumaraguru College of Technology for his valuable support to come out with this project.

I really feel delighted in expressing my heartful thanks to **Prof.Dr.S.Thangasamy, B.E. (Hons)., Ph.D.,** Head of the Department of Computer Science and Engineering for his endless encouragement in carrying out this project successfully.

My heartfelt thanks to our Course Coordinator **Mr.K.R.Baskaran, B.E, M.S.,** for his unfailing enthusiasm, encouragement and guidance that paved me to the completion of this project.

I am indent to express my heartiest thanks to **Mr.R.Rajasehar, M.C.A.,** my project guide who rendered her valuable guidance and support to do this project work extremely well.

I am greatly indebted to C.E.O, Advance Ecom Solutions, Coimbatore for getting me into his esteemed Institution. I also thank **Mr.B.Ramesh, M.C.A.,** who was my guide and he has helped me a lot in my project.

I am also thankful to all the faculty members of the Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore for their valuable guidance, support and encouragement during the course of my project work...

My humble gratitude and thanks to my parents who have supported to complete the project and to my friends for lending me valuable tips, support and cooperation through out my project work.

SYNOPSIS

SYNOPSIS

The goal of the project is to develop an **Intranet And Internet Mail Server** for a software organization. This server is developed to satisfy all the e-mail needs of the organization like sending and receiving emails within the intranet and also in the internet. The mails are shared between the local users in the campus and also been send outside by the users with pop3 rights.

The server is build up with SMTP component to handle the sending of the mail and pop3 component, which handles the receiving of the mail. The existing users of the company's domain, who are all the employees, are registered in the mail server to enable their mailing needs through this server. All the users registered are considered as local users and can send and receive mails between them locally. But some of them are given pop3 rights to mail outside to other domains through Internet. The server is to be developed with various features like anti spam, multi domain support, logging the operations taking place in the server. The effective archiving facility used by the C.E.O to keep track of all the details of the mails sent, received is developed in the mail server. The most security is provided to the content of the data base by providing the back up feature. The front-end tool is developed for the effective management of the user settings and all other settings by the administrator.

Instant messaging facility is also provided with the server. With this facility the users connected in the LAN can chat with each other. In the public room the messages send by all the users are displayed and also a list of online users is maintained. If a user wants to have conversation with other user personally, he can just click on the corresponding username, a private chat window appears in which they can communicate.

CONTENTS

CONTENTS

	CHAPTERS	PAGE NO:
1.	**INTRODUCTION**	01
	1.1 PROJECT OVERVIEW	01
	1.2 ORGANIZATION PROFILE	04
2.	**SYSTEM STUDY AND ANALYSIS**	08
	2.1 EXISTING SYSTEM	08
	2.2 PROPOSED SYSTEM	09
3.	**PROGRAMMING ENVIRONMENT**	12
	3.1 HARDWARE CONFIGURATION	12
	3.2 DESCRIPTION OF SOFTWARE AND TOOLS USED	13
	3.3 PROTOCOLS USED	23
4.	**SYSTEM DESIGN**	32
	4.1 INPUT DESIGN	32
	4.2 PROCESS DESIGN	33
	4.3 TABLE DESIGN	39
5.	**SYSTEM IMPLEMENTATION AND TESTING**	44
	5.1 SYSTEM IMPLEMENTATION	44
	5.2 SYSTEM TESTING	45
6.	**CONCLUSION**	48
7.	**SCOPE FOR FURTHER DEVELOPMENT**	49
8.	**BIBLIOGRAPHY**	50
9.	**APPENDIX**	51
	9.1 SAMPLE SCREENS	51

INTRODUCTION

1.0 INTRODUCTION

1.1 PROJECT OVERVIEW

E-MAILING SYSTEM

E-mail allows impulsiveness of the telephone with the luxury of time and thought that letter writing allows to send and receive E-mail to someone outside the company.

E-mail possesses the list of below properties:

- It is usually fairly short and addresses a single topic.
- It realizes on playing text and relies to previous message.
- It can be sent through one person or many people.

A Server is anything that has some resources that can be shared. Here server is a computer that provides services and resources for other computers. The server also allows processing to be distributed out to the teams so that they can customize applications to meet their particular needs. Basically servers are classified as Dedicated Server and General Purpose Server. And we are dealing with Mail Server, which is a dedicated one.

AN EMAIL CLIENT

An email client is an application that is used to read, write and send email. In simple terms it is the user interface to the email system. The client usually consists of a combination of a simple text editor, address book, filing cabinet and communications module. The text editor allows for the creation of the message itself, and usually includes simple spell checking and formatting facilities. The ability to allow files or documents to be attached to the message is also available.

For example a diagram or schematic could be attached to an email message, offering the recipient the chance to see a project's progress, and comment on it with a reply. The address book allows the users to store commonly used email addresses in an easy to get at format, reducing the chance of addressing errors.

MAIL SERVER:

Mail Servers move mail from client programs through the Internet to its destination hosts and stores it until it is retrieved. A mail server usually consists of a storage area, a set of user definable rules, a list of users and a series of communicational modules. The storage area is where mail is stored for local users, and where messages that are in transit to another destination are temporarily stored. The communication modules are the components that actually handle the transfer of messages to and from other mail servers and e-mail clients.

The aim of the project is to develop an **Intranet And Internet Mail Server** for a software organization. This server is developed to satisfy all the e-mail needs of the organization like sending and receiving emails within the intranet and also in the Internet. The mails are shared between the local users in the campus and also been send outside by the users with pop3 rights.

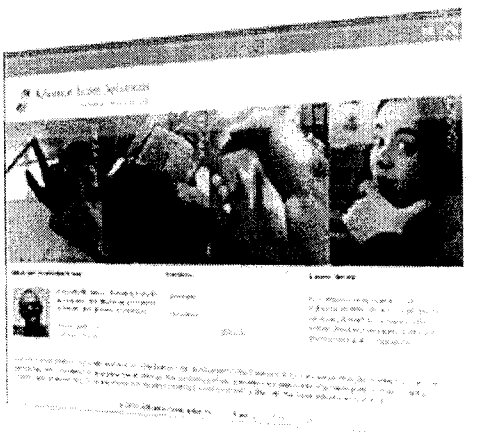
The server is build up with SMTP component to handle the sending of the mail and pop3 component, which handles the receiving of the mail. The existing users of the company's domain, who are all the employees, are registered in the mail server to enable their mailing needs through this server. All the users registered are considered as local users and can send and receive mails between them locally. But some of them are given pop3 rights to mail outside to other domains through Internet.

The server is to be developed with various features like anti spam, multi domain support, logging the operations taking place in the server. The effective archiving facility used by the C.E.O to keep track of the all the details of the mails sent, received is developed in the mail serve. The most security is provided to the content of the data base by providing the back up feature. The front end tool is developed for the effective management of the settings by the administrator.

Instant messaging facility is also provided with the server. With this facility the users connected in the LAN can chat with each other. In the public room the messages send by all the users are displayed and also a list of online users is maintained. If a user wants to have conversation with other user personally, he can just click on the corresponding username, a private chat window appears in which they can communicate.

1.2 ORGANIZATION PROFILE.

Advance Ecom Solutions
Gateway to the future



Advance Ecom Solution (AES) is an offshore web and software development company focused on creating custom applications and laying special emphasis on the unique business needs of its corporate clients.

Our infrastructure and processes are concentrated at designing and implementing solutions for mid-sized companies. AES's long-term-relationship-based strategy ensures a continuous stable growth and a solid customer base: **80% of our orders are repeated orders from our past clients and partners.**

OUR CUSTOMERS:

AES has been partnering with a lot of customers in the US and Western Europe ranging from 25 to 500 employees in size and $1 Million to $50 Million in revenues. AES customers are involved in manufacturing, transport, banking, real estate, tourism, human resources, retail/e-tail, telecommunications and software development industries.

Our customers' success is our success. We're offering innovative web development services that enhance competitive power of our customers. We seek to create and maintain stable long-term relationships based on trust, respect and mutual understanding. We always try to ensure that doing business with AES always proves easy and maximally efficient.

CORPORATE ENVIRONMENT:

The corporate environment at Advance Ecom Solutions promotes and rewards personal responsibility, reliability and collaboration. Internally, we are a team with a shared responsibility for the successes of AES and of our customers. Advance Ecom Solutions is a company managed by owners.

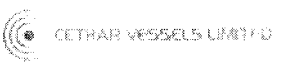
CAPABILITIES:

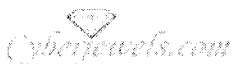
We can broadly classify our experiences under specific type of work we have done as listed below:

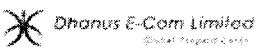
- B2B Info Exchange
- Business Intelligence solutions
- Content Management Systems for in house site management
- Enterprise eCRM solutions – Web based customer support
- Databases driven websites
- Document Management

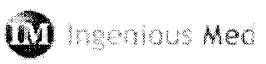
- e-Procurement - Quoting, bidding, price comparison, supplier management and order management
- e-Security – Secured server solutions, Third part encryptions and security deployment
- e-Stores - B2C shopping cart, order, inventory, shipping integration and Merchant account integration with Payment gateway integrations.
- Email Marketing solutions
- Extranets – Corporate extranet, web mail management and corporate portals.
- Mobile & PDA based application
- Online Catalogue Management – Product catalogues with product managements
- Portals – Complete web portals
- Sales Force Automation
- Systems Integration
- IT consulting and solutions findings
- Basic Web Site Design & Development
- Search Engine Optimization
- Web Services

Clients:









KARL KÜBEL INSTITUTE FOR
DEVELOPMENT EDUCATION

Kovai Medical
Center and
Hospital

SYSTEM STUDY AND ANALYSIS

2.0 SYSTEM STUDY AND ANALYSIS

2.1 EXISTING SYSTEM

The Existing system is a pop3 server component, which is used only to receive the mails for the employees in the organization. The sending of the mails is done the users only by directly logging on to the domain's web server. This system does not contain a front end management area for the administrator to do the do the management work.

Any new users can be added only by directly entering into the database, also the deleting, editing of the user details is done by the same way.

This existing component handles the receiving of mails for the users of only a single domain, so the multi domain facility is not available. There is no privacy for the mails of the individual users since the administrator is allowed to enter the data base without any security. so the administrator can read the mails of all the users which may lead to abuse.

Also in the existing component there is no security for the mails since there is no backup facility. The log feature is also not available in the existing system which is used as a guide to monitor the operations done through the server component.

The archive facility, which is used to archive the details of whole mails, that is received for a period of time is not available here. The anti spam used to check the incoming mails for its purpose, size, etc is not present in the existing system. There is no facility for an effective user management by the concern. The content filtering facility is also not available in the existing system.

2.2 PROPOSED SYSTEM

The proposed system is a highly secure, scalable and reliable Intranet Internet Mail Server used to manage and satisfy all the email needs of an organization. It handles the both the sending and receiving mail operation.

This is mainly developed to handle the mail needs of the users of company's domain who all are the employees in the organization. This server is built with the SMTP component used for sending mail and POP3 component used for receiving the mail. Both the components are developed using the rules of the SMTP and POP3 protocols.

The server is a transmitter designed to handle the mail needs of only the existing users with an e-mail account in the domain. It is not a web sever to create a new user, giving an e-mail account, etc. It handles the mail needs of the users who are all registered in the server.

This server is developed to have two-user settings i.e. local user setting and pop3 user setting. Local users are users who are able to send and receive mails within the organization and are not allowed to send or receive mail from outside. The pop3 users are local users who had enabled the pop3 facility. All the employees are local users. But some of them are given the pop3 rights to send mails outside and receive from outside.

The proposed system overcomes all the draw backs of the existing system .it has an effective front end management area which is used by the administrator to do all the management purposes like adding new users, deleting users and also to do the necessary settings in the following facilities. The mails of the individual persons are secure and are not able to view by the administrator.

The client side operations to send or receive a mail are done by using an existing e-mail client like outlook express. It is used to read mail, compose new mail and maintaining the address book. This server is developed effectively with the following facilities.

MULTI DOMAIN SUPPORT:

The server is designed not only to handle the mail needs of users of single domain. But also it has the facility of handling mail of multiple domains which is related to the organization.

ANTI SPAM:

The server is provided with an effective anti spam facility with a set of rules to declare a mail as spam free or spam and do some actions accordingly. The filtration is done based on the domain name, e-mail id, size of the mail and the inbox count. The actions are block the mail or send the mail to the administrator

CONTENT FILTERING:

The content filter used to filter the contents of the incoming mails for some unwanted words, expressions in the mail and do necessary actions is provided in the proposed system.

LOGGING:

The proposed system has an effective logging feature which is used as a template to track all the actions done in the server. It has inbox log to display the actions done in inbox, out box log to display the actions done in outbox and a general log to track all the operations done in the mail server. The displayed log is will be written into a text file for the future reference.

ARCHIVING:

The mail server is provided with an excellent achieve feature. This is used to archive the contents of the mails send and received by the employees and write it as a text file for the reference of higher authorities in the company. This feature cannot be used by the administrator with out the permission of the authority. Hence the privacy of individual persons cannot be affected.

BACK UP:

The proposed system is provided with the facility of backing up of all the necessary details present in the data base. This can be exported as a text file in the comma separated format into a secondary storage device which may be imported and used when required.

INSTANT MESSAGING:

The proposed system has an additional feature of instant messaging; with this facility the users connected in the LAN can chat with each other. In the public room the messages send by all the users are displayed and also a list of online users is maintained. If a user wants to have conversation with other user personally, he can just click on the corresponding username, a private chat window appears in which they can communicate.

PROGRAMMING ENVIRONMENT

3.0 PROGRAMMING ENVIRONMENT

3.1 HARDWARE CONFIGURATION

Processor	:	Pentium III
RAM	:	128 MB
Hard Disk	:	10GB
Monitor	:	VGA Color Monitor
Network Card (Speed)	:	100 Mbps
Network	:	any supporting TCP/IP

3.2 DESCRIPTION OF SOFTWARE'S AND TOOLS USED

MICROSOFT .NET FRAMEWORK:

The Microsoft .NET Framework is a platform for building, deploying, and running Web Services and applications. It provides a highly productive, standards-based, multi-language environment for integrating existing investments with next-generation applications and services as well as the agility to solve the challenges of deployment and operation of Internet-scale applications. The .NET Framework consists of three main parts: the common language runtime, a hierarchical set of unified class libraries, and a componentized version of Active Server Pages called ASP.NET.

COMMON LANGUAGE RUNTIME (CLR)

The common language runtime is the execution engine for .NET Framework applications.

It provides a number of services, including the following:

- Code management (loading and execution)
- Application memory isolation
- Verification of type safety
- Conversion of IL to native code
- Access to metadata (enhanced type information)
- Managing memory for managed objects
- Enforcement of code access security
- Exception handling, including cross-language exceptions
- Interoperation between managed code, COM objects, and pre-existing DLLs (unmanaged code and data)

- Automation of object layout
- Support for developer services (profiling, debugging, and so on)

COMMON TYPE SYSTEM (CTS)

The common type system is a rich type system, built into the common language runtime that supports the types and operations found in most programming languages. The common type system supports the complete implementation of a wide range of programming languages.

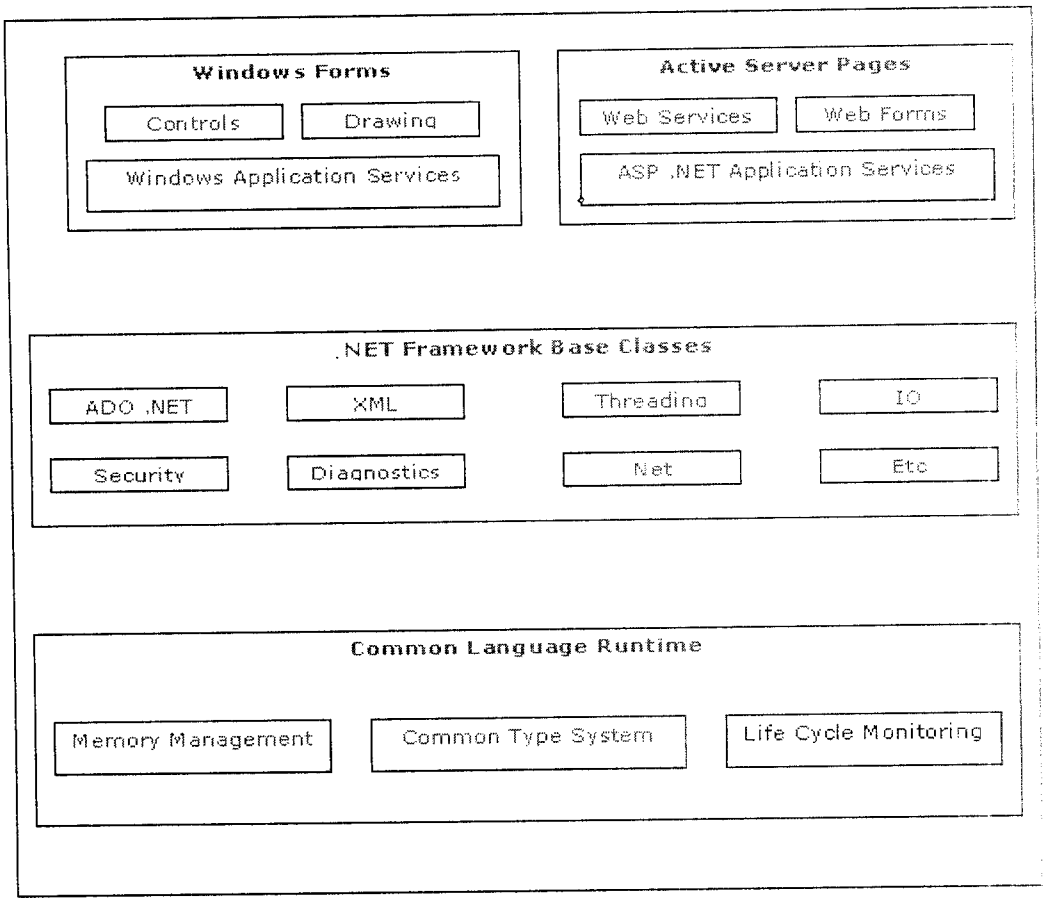
COMMON LANGUAGE SPECIFICATION (CLS)

The Common Language Specification is a set of constructs and constraints that serves as a guide for library writers and compiler writers. It allows libraries to be fully usable from any language supporting the CLS, and for those languages to integrate with each other. The Common Language Specification is a subset of the common type system. The Common Language Specification is also important to application developers who are writing code that will be used by other developers. When developers design publicly accessible APIs following the rules of the CLS, those APIs are easily used from all other programming languages that target the common language runtime.

MICROSOFT INTERMEDIATE LANGUAGE (MSIL)

MSIL is the CPU-independent instruction set into which .NET Framework programs are compiled. It contains instructions for loading, storing, initializing, and calling methods on objects.

Combined with metadata and the common type system, MSIL allows for true cross-language integration. Prior to execution, MSIL is converted to machine code. It is not interpreted.



.NET Framework

ASSEMBLY:

An assembly is the primary building block of a .NET Framework application. It is a collection of functionality that is built, versioned, and deployed as a single implementation unit (as one or more files). All managed types and resources are marked either as accessible only within their implementation unit, or as accessible by code outside that unit.

Assemblies are self-describing by means of their manifest, which is an integral part of every assembly. The manifest:

- Establishes the assembly identity (in the form of a text name), version, culture, and digital signature (if the assembly is to be shared across applications).
- Defines what files (by name and file hash) make up the assembly implementation.
- Specifies the types and resources that make up the assembly, including which are exported from the assembly.
- Itemizes the compile-time dependencies on other assemblies.
- Specifies the set of permissions required for the assembly to run properly.

This information is used at run time to resolve references, enforce version binding policy, and validate the integrity of loaded assemblies. The runtime can determine and locate the assembly for any running object, since every type is loaded in the context of an assembly. Assemblies are also the unit at which code access security permissions are applied. The identity evidence for each assembly is considered separately when determining what permissions to grant the code it contains.

The self-describing nature of assemblies also helps makes zero-impact install and XCOPY deployment feasible.

VISUAL BASIC.NET

Visual Basic .NET, the next generation of the Visual Basic language, is designed to be a fast and easy way to create .NET applications, including XML Web services and ASP.NET Web applications.

Visual Basic .NET has many new and improved features such as inheritance, interfaces, and overloading that make it a powerful object-oriented programming language. Other new language features include free threading and structured exception handling. Visual Basic .NET fully integrates the .NET Framework and the Common Language Runtime, which together provide

language interoperability, garbage collection, enhanced security, and improved versioning support.

Visual Basic .NET has many new and improved language features — such as inheritance, interfaces, and overloading — that make it a powerful object-oriented programming language. As a Visual Basic developer, you can now create multithreaded, scalable applications using explicit multithreading. Other new language features in Visual Basic .NET include structured exception handling, custom attributes, and common language specification (CLS) compliance.

BIT SHIFT OPERATORS

Visual Basic .NET now supports arithmetic left and right shift operations on integral data types (**Byte**, **Short**, **Integer**, and **Long**). Arithmetic shifts are not circular, which means the bits shifted off one end of the result are not reintroduced at the other end. The corresponding assignment operators are provided as well. For details, see Bit Shift Operators and Assignment Operators.

LOOP VARIABLE DECLARATION

Visual Basic .NET now allows you to declare a loop variable as part of a **For** or **For Each** loop. You can include an **As** clause for the variable in the **For** or **For Each** statement, provided no variable of that name has been declared outside the loop. The scope of a loop variable declared in this manner is the loop itself. For details, see for...Next Statements and for each...Next Statements.

INHERITANCE

Visual Basic .NET supports *inheritance* by allowing you to define classes that serve as the basis for derived classes. Derived classes inherit and can extend the properties and methods of the base class. They can also override inherited methods with new implementations. All classes created with Visual Basic .NET are inheritable by default. Because the forms you design are really classes, you can use inheritance to define new forms based on existing ones. For details, see Inheritance.

EXCEPTION HANDLING

Visual Basic .NET supports *structured exception handling*, using an enhanced version of the **Try...Catch...Finally** syntax supported by other languages such as C++. Structured exception handling combines a modern control structure (similar to **Select Case** or **While**) with exceptions, protected blocks of code, and filters. Structured exception handling makes it easy to create and maintain programs with robust, comprehensive error handlers. For details, see Exception Handling.

OVERLOADING

Overloading is the ability to define properties, methods, or procedures that have the same name but use different data types. Overloaded procedures allow you to provide as many implementations as necessary to handle different kinds of data, while giving the appearance of a single, versatile procedure. For details, see Overloaded Properties and Methods.

OVERRIDING PROPERTIES AND METHODS

The **Overrides** keyword allows derived objects to override characteristics inherited from parent objects. Overridden members have the same arguments as the members inherited from the base class, but different implementations. A member's new implementation can call the original implementation in the parent class by preceding the member name with **My Base**. For details, see Overriding Properties and Methods.

CONSTRUCTORS AND DESTRUCTORS

Constructors are procedures that control initialization of new instances of a class. Conversely, *destructors* are methods that free system resources when a class leaves scope or is set to **nothing.** Visual Basic .NET supports constructors and destructors using the **Sub New** and **Sub Finalize** procedures. For details, see Object Lifetime: How Objects are Created and Destroyed.

DATA TYPES

Visual Basic .NET introduces three new data types. The **Char** data type is an unsigned 16-bit quantity used to store Unicode characters. It is equivalent to the .NET Framework **System. Char** data type. The **Short** data type, a signed 16-bit integer, was named **Integer** in earlier versions of Visual Basic. The **Decimal** data type is a 96-bit signed integer scaled by a variable power of 10. In earlier versions of Visual Basic, it was available only within a **Variant**. For details, see Data Types.

INTERFACES

Interfaces describe the properties and methods of classes, but unlike classes, do not provide implementations. The Interface statement allows you to declare interfaces, while the Implements statement lets you write code that puts the items described in the interface into practice. For details, see Interfaces in Visual Basic .NET.

DELEGATES

Delegates — objects that can call the methods of objects on your behalf — are sometimes described as type-safe, object-oriented function pointers. You can use delegates to let procedures specify an event handler method that runs when an event occurs. You can also use delegates with multithreaded applications. For details, see Delegates and the Address of Operator.

MULTITHREADING

Visual Basic .NET allows you to write applications that can perform multiple tasks independently. A task that has the potential of holding up other tasks can execute on a separate thread, a process known as multithreading. By causing complicated tasks to run on threads that are separate from your user interface, multithreading makes your applications more responsive to user input. For details, see Multithreaded Applications.

MS-ACCESS:

A BRIEF ACCESS HISTORY

Access 1.0 really opened the eyes of many database developers. It was one of the first relational database products available for the Windows 3 platform, and it was certainly the first to fill the needs of many developers, both corporate and independent. Access 1.0 made it very easy to create simple applications. It did have some limitations when developers got past a certain point in their applications, and it had a severe limitation in that databases couldn't be larger than 128 megabytes. Access 1.1 fixed that limitation, expanding the maximum database size to 1 gigabyte, and fixed some other limitations as well.

Access 2.0 offered great gains for developers. Although it also provided numerous improvements for end users, the greatest leap from 1.1 came in the improvement for the developer community. For the professional programmer Access2.0 added features in almost every area of the product.

Features added in Access 2.0

- A vastly extended object and event model
- Run-time access to most form and report properties
- Event procedures
- Cascading updates and deletes for referential integrity

- Rushmore query optimization
- Data access objects (DAO), a consistent object model for the manipulation of Jet engine data
- Support for 32-bit OLE custom controls

The Access Basic language and integrated development environment (IDE) where replaced with Visual Basic for Applications (VBA) and its enhanced IDE.

Access 2003 was the next release with minor comparison to Access 2000. With Access 2003 further enhancements were made in the following areas:

- Internet and Intranet features
- The VBA integrated development environment
- Shared Microsoft Office programmability features
- Data Access Objects

ACCESS 2003

Access is the best-selling desktop database program on the market today. It has the right mix of features for both users and developers. Microsoft created this large user base by designing Microsoft Access to be a single database tool that all type of users could use-from database novices doing their first query to database developers building optimized inner-join double back flips.

Microsoft Access is the standard against which other tools are measured for ease of use and database development. Access 2003 is a wonderfully powerful development platform.

FEATURES

Microsoft has focused on making Access more Internet / intranet friendly. A host of features has been added, including these:

- Tables have a Hyperlink data type
- The Publish to the Web wizard makes it easy to publish data on the internet or your corporate intranet
- We can export tables and queries to HTML
- We can insert the Web browser control into any Access form

3.3 PROTOCOLS USED

SMTP PROTOCOL
SIMPLE MAIL TRANSFER PROTOCOL

INTRODUCTION:

The objective of Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently.

SMTP is independent of the particular transmission subsystem and requires only a reliable ordered data stream channel. Appendices A, B, C, and D describe the use of SMTP with various transport services. A Glossary provides the definitions of terms as used in this document. An important feature of SMTP is its capability to relay mail across
transport service environments. A transport service provides process communication environment (IPCE). An IPCE may cover one network, several networks, or a subset of a network. It is important to realize that transport systems (or IPCEs) are not one-to-one with networks. A process can communicate directly with another process through any mutually known IPCE. Mail is an application or use of inter process communication. Mail can be communicated between processes in different IPCEs by relaying through a process connected to two (or more) IPCEs. More specifically, mail can be relayed between hosts on different transport systems by a host on both transport systems.

The SMTP model

The SMTP design is based on the following model of communication: as the result of a user mail request, the sender-SMTP establishes a two-way transmission channel to a receiver-SMTP. The receiver-SMTP may be either the ultimate destination or an intermediate. SMTP commands are generated by the

sender-SMTP and sent to the receiver-SMTP. SMTP replies are sent from the receiver-SMTP to the sender-SMTP in response to the commands.

Once the transmission channel is established, the SMTP-sender sends a MAIL command indicating the sender of the mail. If the SMTP-receiver can accept mail it responds with an OK reply. The SMTP-sender then sends a RCPT command identifying a recipient of the mail. If the SMTP-receiver can accept mail for that recipient it responds with an OK reply; if not, it responds with a reply rejecting that recipient (but not the whole mail transaction). The SMTP-sender and SMTP-receiver may negotiate several recipients. When the recipients have been negotiated the SMTP-sender sends the mail data, terminating with a special sequence. If the SMTP-receiver successfully processes the mail data it responds with an OK reply. The dialog is purposely lock-step, one-at-a-time.

COMMAND SEMANTICS

The SMTP commands define the mail transfer or the mail system function requested by the user. SMTP commands are character strings terminated by <CRLF>.

HELLO (HELO)

This command is used to identify the sender-SMTP to the receiver-SMTP. The argument field contains the host name of the sender-SMTP. The receiver-SMTP identifies itself to the sender-SMTP in the connection greeting reply, and in the response to this
command.

MAIL (MAIL)

This command is used to initiate a mail transaction in which the mail data is delivered to one or more mailboxes. The argument field contains a reverse-path.

RECIPIENT (RCPT)

This command is used to identify an individual recipient of the mail data; multiple recipients are specified by multiple use of this command.

DATA (DATA)

The receiver treats the lines following the command as mail data from the sender. This command causes the mail data from this command to be appended to the mail data buffer. The mail data may contain any of the 128 ASCII character codes.

The mail data is terminated by a line containing only a period, that is the character sequence "<CRLF>.<CRLF>" (see Section 4.5.2 on Transparency). This is the end of mail data indication.

SEND (SEND)

This command is used to initiate a mail transaction in which the mail data is delivered to one or more terminals. The argument field contains a reverse-path. This command is successful if the message is delivered to a terminal.

SEND AND MAIL (SAML)

This command is used to initiate a mail transaction in which the mail data is delivered to one or more terminals and mailboxes. For each recipient the mail data is delivered to the recipient's terminal if the recipient is active on the host (and accepting terminal messages), and for all recipients to the recipient's mailbox. The argument field contains a reverse-path. This command is successful if the message is delivered to the mailbox.

RESET (RSET)

This command specifies that the current mail transaction is to be aborted. Any stored sender, recipients, and mail data must be discarded, and all buffers and state tables cleared. The receiver must send an OK reply.

VERIFY (VRFY)

This command asks the receiver to confirm that the argument identifies a user. If it is a user name, the full name of the user (if known) and the fully specified mailbox are returned. This command has no effect on any of the reverse-path buffer, the forward-path buffer, or the mail data buffer. EXPAND (EXPN). This command asks the receiver to confirm that the argument identifies a mailing list, and if so, to return the membership of that list. The full name of the users (if known) and the fully specified mailboxes are returned in a multi line reply. This command has no effect on any of the reverse-path buffer, the forward-path buffer, or the mail data buffer.

HELP (HELP)

This command causes the receiver to send helpful information to the sender of the HELP command. The command may take an argument (e.g., any command name) and return more specific information as a response. This command has no effect on any of the reverse-path buffer, the forward-path buffer, or the mail data buffer.

NOOP (NOOP)

This command does not affect any parameters or previously entered commands. It specifies no action other than that the receiver send an OK reply. This command has no effect on any of the reverse-path buffer, the forward-path buffer, or the mail data buffer.

QUIT (QUIT)

This command specifies that the receiver must send an OK reply, and then close the transmission channel. The receiver should not close the transmission channel until it receives and replies to a QUIT command (even if there was an error). The sender should not close the transmission channel until it sends a QUIT command and receives the reply (even if there was an error response to a previous command). If the connection is closed prematurely the receiver should act as if a RSET command had been received (canceling any pending transaction, but not undoing any previously completed transaction), the sender should act as if the command or transaction in progress had received a temporary error (4xx).

SMTP replies

Replies to SMTP commands are devised to ensure the synchronization of requests and actions in the process of mail transfer, and to guarantee that the sender-SMTP always knows the state of the receiver-SMTP. Every command must generate exactly one reply.

The details of the command-reply sequence are made explicit in Section 5.3 on Sequencing and Section 5.4 State Diagrams. An SMTP reply consists of a three digit number (transmitted as three alphanumeric characters) followed by some text. The number is intended for use by automata to determine what state to enter next; the text is meant for the human user. It is intended that the three digits contain enough encoded information that the sender-SMTP need not examine the text and may either discard it or pass it on to the user, as appropriate. However multi line replies are allowed for any command.

Reply codes by function groups

500 Syntax error, command unrecognized [This may include errors such as command line too long]
501 Syntax errors in parameters or arguments
502 Command not implemented
503 Bad sequence of commands
504 Command parameter not implemented

211 System status or system help reply
214 Help message
[Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user]

220 <domain> Service ready
221 <domain> Service closing transmission channels
421 <domain> Service not available,
Closing transmission channel [This may be a reply to any command if the service knows it must shut down]
250 Requested mail action okay, completed
251 User not local; will forward to <forward-path>
450 Requested mail actions not taken: mailbox unavailable
[E.g., mailbox busy]
550 Requested actions not taken: mailbox unavailable
[E.g., mailbox not found, no access]
451 Requested actions aborted: error in processing
551 User not local; please try <forward-path>
452 Requested actions not taken: insufficient system storage
354 Start mail input; end with <CRLF>.<CRLF>
554 Transaction failed

POP3 PROTOCOL

POST OFFICE PROTOCOL

INTRODUCTION:

On certain types of smaller nodes in the Internet it is often impractical to maintain a message transport system (MTS). For example, a workstation may not have sufficient resources (cycles, disk space) in order to permit a SMTP server [RFC821] and associated local mail delivery system to be kept resident and continuously running. Similarly, it may be expensive (or impossible) to keep a personal computer interconnected to an IP-style network for long amounts of time (the node is lacking the resource known as "connectivity").

Despite this, it is often very useful to be able to manage mail on these smaller nodes, and they often support a user agent (UA) to aid the tasks of mail handling. To solve this problem, a node which can support an MTS entity offers a mail drop service to these less endowed nodes. The Post Office Protocol Version 3 (POP3) is intended to permit a workstation to dynamically access a mail drop on a server host in a useful fashion. Usually, this means that the POP3 protocol is used to allow a workstation to retrieve mail that the server is holding for it. POP3 is not intended to provide extensive manipulation operations of mail on the server; normally, mail is downloaded and then deleted.

BASIC OPERATION:

Initially, the server host starts the POP3 service by listening on TCP port 110. When a client host wishes to make use of the service, it establishes a TCP connection with the server host. When the connection is established, the POP3 server sends a greeting. The client and POP3 server then exchange commands and responses (respectively) until the connection is closed or aborted. Commands in the POP3 consist of a case-insensitive keyword, possibly followed by one or more arguments. All commands are terminated by a CRLF pair. Keywords and arguments consist of printable ASCII characters. Keywords and arguments are

each separated by a single SPACE character. Keywords are three or four characters long. Each argument may be up to 40 characters long. Responses in the POP3 consist of a status indicator and a keyword possibly followed by additional information. All responses are terminated by a CRLF pair. Responses may be up to 512 characters long, including the terminating CRLF. There are currently two status indicators: positive ("+OK") and negative ("-ERR"). Servers MUST send the "+OK" and "-ERR" in upper case.

THE AUTHORIZATION STATE

In this state, the POP3 client sends the user name and password to the POP3 server for the authentication. If the user successfully authenticated, the next stage transaction is entered. Otherwise, the server disconnects the client.

THE TRANSACTION STATE

Once the client has successfully identified itself to the POP3 server and POP3 server has looked and opened the mailbox, the POP3 session is in the TRANSACTION state. The client may now issue POP3 commands repeatedly to list, retrieve and delete e-mail messages in the mail box.

THE UPDATED STATE

Upon completion of the commands in the TRANSACTION state, the client can issue the QUIT command to end the session and place the POP3 server in the UPDATE state. If the client should issue the QUIT command from AUTHORIZATION state, the POP3 session terminates but does to enter the UPDATE state. In the UPDATE state, the POP3 server removes all messages marked for deletion from the mailbox.

STAT

Retrieves the status of the mailbox with the authorized user. The returned status message includes the number of messages in the mailbox and the accumulated size of all messages.

LIST

List all the e-mail messages in the receipt order, with a message number and size for each message.

RETR msg

Retrieves the message designated by the number in msg.

DELE msg

Deletes the message designated by the number in msg.

RSET

Unmark all messages marked for deletion by the DELE command.

TOP msg

Provides the top n lines of the body of the message designated by the number in msg.

UIDL [msg]

Retrieves the unique ID Listing number for all messages or the messages specified by the number in msg.

TCP PROTOCOL

The Transmission Control Protocol is one of the main protocols in TCP/IP networks used to establish a strong connection between two systems wanted to exchange data with each other. TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent.

SYSTEM DESIGN

4.0 SYSTEM DESIGN

4.1 INPUT DESIGN

Input design is the most integral part of any application and this is accomplished with the help of user input screens. The input data has to be edited, validated, organized, and accepted by the system before being processed to produce the outputs

Here in the mail server the input are entered by two ways for processing of mail and for the management activities.

ADMINISTRATOR INPUT

In our mail server the administrator uses out front end user interface to enter the data regarding the management purposes. The login inputs are entered in the first stage. The registration of a new user with all their details, deletion, editing are given as the input by the administrator.

The details of the domains, adding, deleting the domains are also inputted. The other inputs are entries in the anti spam process, content filtering details. The logging file entries are given and also the back up process entries are inputted.

Only for the archiving process the administrator doesn't have rights to enter, so the input is given with the permission of the C.E.O of the concern.

USER INPUT

The users have to enter their username and password as the input in the mail client; they compose mail which will be then used in sending and receiving of mail. The same is required for logging into the instant messaging and then the available users have conversation with each other.

4.2 PROCESS DESIGN

The process of the mail server includes three main modules, the mail sending, mail receiving and the instant messaging. The mail sending process is done by the SMTP server and the mail receiving process is done by the POP3 server. The instant messaging is done by the IM server.

THE SMTP SERVER:

The mail sending process is done mainly in three stages.

a. Remote system identification

b. Connection establishment

c. Transferring of data

The SMTP server acts in two ways both as a server and as a client in process of sending the mails. First the SMTP server acts as a server with the server commands

1. The e-mail client requests for a connection to the SMTP server in port 25. then the SMTP server responses with the server commands.

2. After the connection has been established, the e-mail client has a conversation with the SMTP server, telling the SMTP server the address of the sender and the address of the recipient, as well as the body of the message.

3. The SMTP server takes the "to" address (jsmith@mindspring.com) and breaks it into two parts:

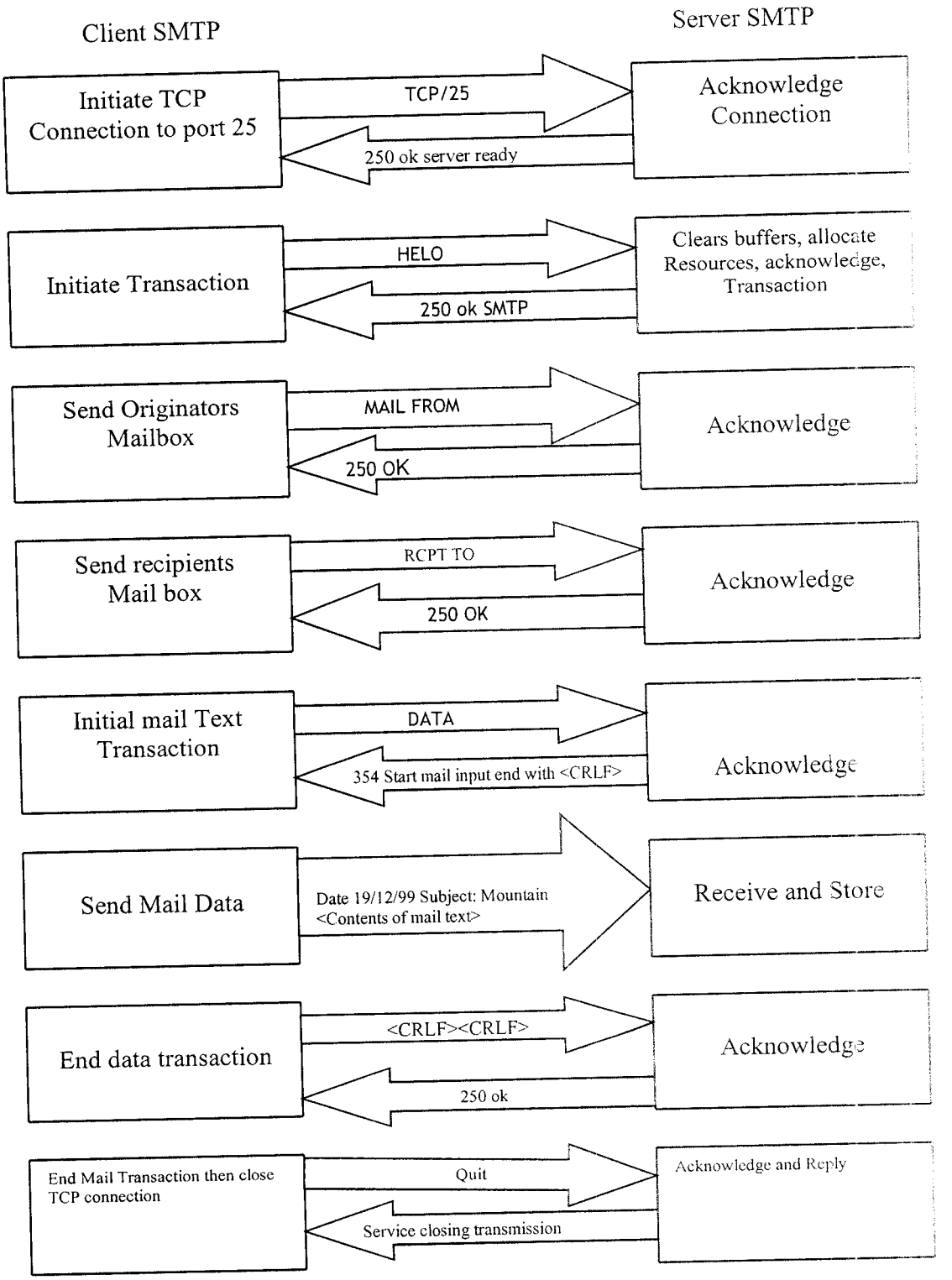
4. Then the server verifies the destination address with the local user database .if it matches then the server will forward the mail to the inbox area of the pop3 server. But if the destination address is not a local address then the server does the following.

After this the SMTP server closes connection with the e-mail client and then starts connecting to the destination server using client commands.

5. The SMTP server has a conversation with a Domain Name Server, or DNS using the MX records components. It says, "Can you give me the IP address of the SMTP server for mindspring.com?" The DNS replies with the one or more IP addresses for the destination SMTP server

6. The SMTP server connects with the destination SMTP server using port 25. It has the same simple text conversation that e-mail client had with the SMTP server and gives the message to the destination server. The destination server recognizes that the domain name for E=mail id is at their user list. so it hands the message to destination's POP3 server, which puts the message in user's mailbox.

The following process diagram explains the detailed conversation of the e-mail client with the SMTP server. The similar conversation is occurred between our SMTP server and the destination server provided our SMTP server uses the client commands.


Client SMTP
Server SMTP
Initiate TCP Connection to port 25
TCP/25
250 ok server ready
Acknowledge Connection
Initiate Transaction
HELO
250 ok SMTP
Clears buffers, allocate Resources, acknowledge, Transaction
Send Originators Mailbox
MAIL FROM
250 OK
Acknowledge
Send recipients Mail box
RCPT TO
250 OK
Acknowledge
Initial mail Text Transaction
DATA
354 Start mail input end with <CRLF>
Acknowledge
Send Mail Data
Date 19/12/99 Subject: Mountain
<Contents of mail text>
Receive and Store
End data transaction
<CRLF><CRLF>
250 ok
Acknowledge
End Mail Transaction then close TCP connection
Quit
Service closing transmission
Acknowledge and Reply

THE POP3 SERVER

The mail receiving process is done in three stages

a. authentication state

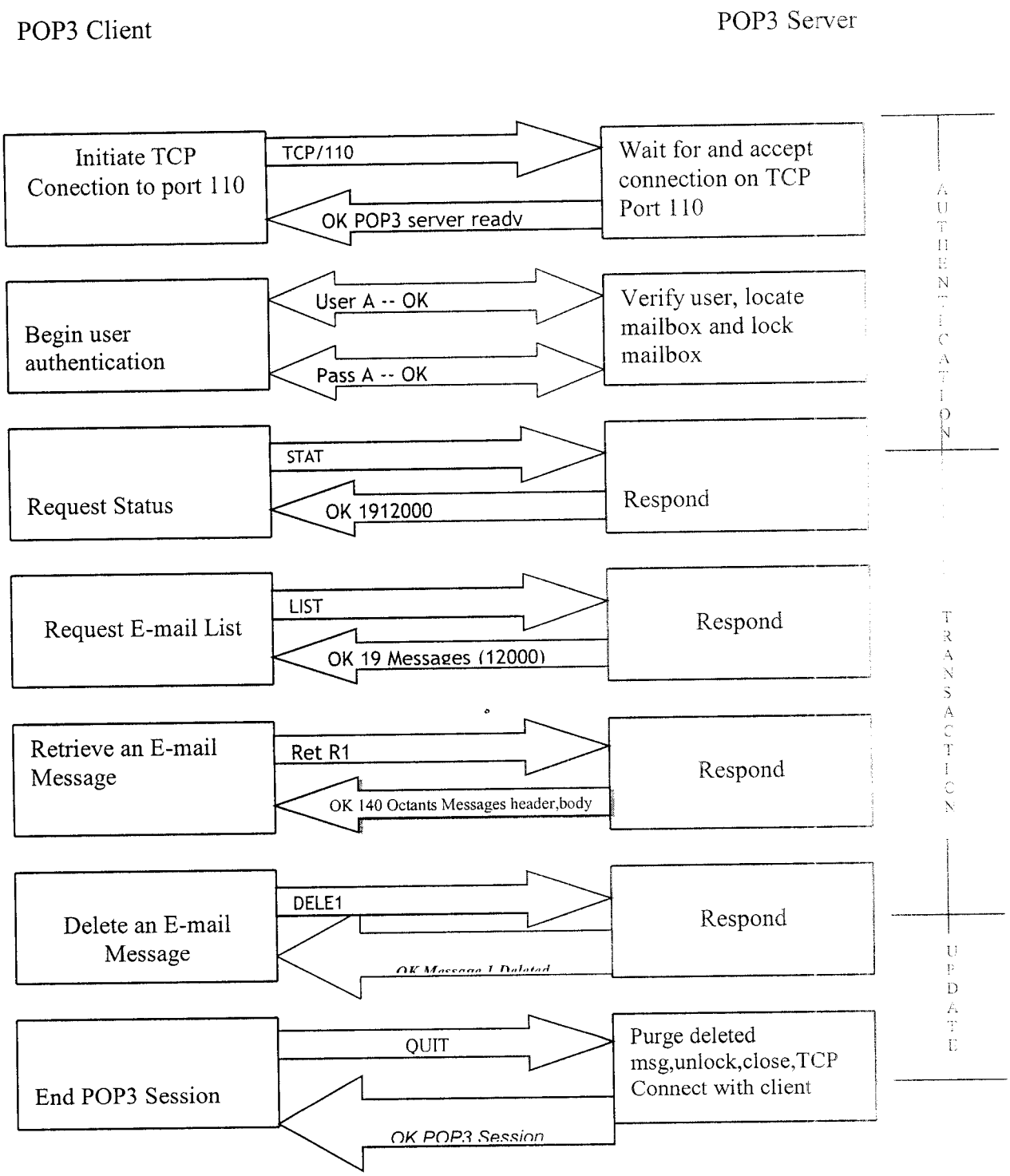
b. transaction state

c. updating state

Similar to the SMTP the POP3 server also acts in two stage as a client to retrieve the mails from the host domains pop3 server and as a server in sending the mails to the local e-mail client. First our POP3 server act as a client with client commands and the following is done to retrieve the mails.

- The POP3 server requests for a connection to the host domain pop3 server which is the source in port 110, then the source server responses with the server commands.
- After the connection has been established, the POP3 server has a conversation with the source server, telling the source server the username and password.
- Then the source server authenticates the username and password in its database. if it is o.k. ,it sends the details of the mails to the POP3 server
- After receiving the mail from the mail the POP3 server closes the connection with the source server. Then the similar above process takes place between POP3 server and the e-mail client to retrieve the mail from POP3 server to the inbox of individual user. Here the POP3 server acts as a server with server commands.

The following process diagram gives the detailed conversation of the POP3 server with the source pop3 server. The same is between the e-mail client and the POP3 server


POP3 Client
POP3 Server
Initiate TCP Conection to port 110
TCP/110
OK POP3 server readv
Wait for and accept connection on TCP Port 110
AUTHENTICATION
Begin user authentication
User A -- OK
Pass A -- OK
Verify user, locate mailbox and lock mailbox
Request Status
STAT
OK 1912000
Respond
Request E-mail List
LIST
OK 19 Messages (12000)
Respond
TRANSACTION
Retrieve an E-mail Message
Ret R1
OK 140 Octants Messages header,body
Respond
Delete an E-mail Message
DELE1
OK Message 1 Deleted
Respond
UPDATE
End POP3 Session
QUIT
OK POP3 Session
Purge deleted msg,unlock,close,TCP Connect with client

IM SERVER

The IM server does the instant messaging process. Mail server waits for the connections from the clients in port 1100. If any client asks for the connection while a user log on, the server updates the database and sends the current user list who all are online.

If the user uses the public chat room and sends some message, the server will deliver the message to all the client systems which are connected to it.

But if the user A selects a particular user B then the server will open a private window and sends the message to that particular client machine.

4.3 TABLE DESIGN

User Master

Name	Data Type	Description
user_id	Number	Identification number for a user
Name	Text	Name of the user
Email_id	Number	Mail address of the user
Password	Text	Password of the user
Department	Text	Department which the user belongs to
Details	Text	The comments about the user
Domain_id	Number	Domain no in which the user have an account
Pop3user	Y/N	Did the user have pop3 rights
Type	Text	admin/CEO/ordinary
Online	Y/N	Whether users online or not
Active	Y/N	Users active or inactive

Domain Master

Name	Data Type	Description
Domain_id	Number	Domain number
Domain_name	Text	Name of Domain

Domain_smtp	Text	SMTP address of the domain
Domain_pop3	Text	POP3 address of the domain
Active	Y/N	Domain active or not

Inbox Details

Name	Data Type	Description
Inbox_id	Number	Inbox number
User_id	Number	User number
Mail_from	Text	From address
Mail_to	Text	To address
Mail_date	Text	Date of the mail
Mail_data	Text	Body of the mail
Status	Y/N	read/new

Outbox Details

Name	Data Type	Description
Outbox_id	Number	Outbox number
User_id	Number	User number
Mail_from	Text	From address

Mail_to	Text	To address
Mail_date	Text	Date of the mail send
Mail_data	Text	Body of the mail

Scheduling Details

Name	Data Type	Description
Time_interval	Number	Time between every operation
Send	Y/N	To enable sending
Receive	Y/N	To enable receiving
Delete_value	Number	Delete the mail after the given number
Order	Text	Weather download according to size

Log Details

Name	Data Type	Deseription
Log_path	Text	Path of file to which the log is stored
Inbox	Y/N	To enable inbox log
Outbox	Y/N	To enable outbox log

Errors	Y/N	To enable error log
User_change_details	Y/N	To enable change user log

Spam Details

Name	Data Type	Description
Block_type	Text	domain or email or size or inbox_count
Blocked_value_action	Text	mail to admin & block,only block

Content Details

Name	Data Type	Description
Content	Text	Words to be filtered
Action	Text	mail to admin & block, only block, remove the content only

Settings

Name	Data Type	Description
Setting_name	Text	Archive_Days, Archive_incoming, Archive_Outgoing
Setting_value	Text	Value of the above settings

SYSTEM IMPLEMENTATION AND TESTING

5.0 System implementation and Testing

5.1 SYSTEM IMPLEMENTATION

- Mail server creates TCP connection in 25 SMTP port and 110 POP3 port to serve the local users.
- Local users connect to the mail server using outlook express
- When the data arrives from the client, it will be forwarded to the SMTP component and the POP3 component
- The result will be send to the client using the TCP connection
- When it connects to the Internet, it will communicate to the outside world using the same port, but as a client.
- It sends the details of the mail using the SMTP port and the Pop3 port receives the mails and stores in the database
- It uses text files to store the mails
- Instant massaging done by server at port1100
- The backup files are written in comma separated format
- Uses ADO.NET connection to connect with the database
- Uses timer control to run in a particular interval
- Uses Winsock hybrid in Winsock library for creating the TCP connection

5.2 SYSTEM TESTING

INTRODUCTION

Software testing is the process of checking whether the developed system is working according to the original objectives and requirements. The system should be tested experimentally with test data so as to ensure that the system works according to the required specification. When the system is found working, test it with actual data and check performance.

Testing presents an interesting anomaly for software engineers. The engineer creates a series of test cases that are intended to demolish the software that has been built. Testing requires that the developer discards preconceived notions "correctness" of software just developed and overcome a conflict of interest that occurs when errors are uncovered.

TESTING OBJECTIVE

The testing objectives are summarized in the following steps:

1) Testing is a process of executing a program with the intent of finding an error.
2) A good test case is one that has high probability of finding an as-yet undiscovered error.
3) A successful test is one that uncovers an as yet undiscovered error.

TESTING PRINCIPLES

1) All tests should be traceable to customer requirement.
2) Tests should be planned long before testing begins that is the test-planning can being as soon as the requirement model is complete.
3) The Pareto principle applies to software testing. The Pareto principle implies that 80 percent of all errors uncovered during

testing will likely be traceable to 20 percent of all program modules.

4) Exhaustive testing is not possible.

5) To be more effective, testing has highest probability of finding errors.

The following are the attributes of a good test:

1. A good test has a high probability of finding an error.
2. A good test is not redundant.
3. A good test should be "best of breed".
4. A good test should be neither too simple nor too complex.

TYPES OF TESTS

Unit Testing

Unit test comprises of a set of task performed by an individual programmer prior to integration of the unit into larger systems. Program unit is usually small enough that the programmer who developed it can test in great detail.

Module unit testing should be as exhaustive as possible to ensure that each representative are handled by each module that has been tested. All the units that make up the system were tested independently to ensure that they work as required.

Integration Testing

Integration Testing is a systematic technique for constructing the program structure at the same time conducting test to uncover errors associated with interfacing. The objective is to take tested module and build a program structure that has been dictated by design.

Bottom-up integration is a traditional strategy used to integrate the component of system software into a functional whole while top-down integration method is an incremental approach to the construction of program structure.

Validation Testing

At the end of integration testing, the software is completely assembled as a package, interfacing errors have been uncovered and correction testing begin.

Validation Test Criteria

Software testing and validation is achieved through a series of Black box tests that demonstrates conformity with the requirements. A test plan outlines the classes of test to be conducted and a test procedure defines specific tests that checks whether the systems requirements are met. Both the plan and procedure are designed to ensure that all functional requirements are achieved.

Acceptance testing

Acceptance test involves planning an execution of functional test and performance tests in order to demonstrate that the implemented software satisfy its requirements. The procedure will incorporate test case developed during unit testing and integration testing. Additional test cases are added to ensure that the system does meet the desired level of function and performance.

Before integration of unit into larger system, each unit is tested to make sure that they work as needed. After assembling the software into package, validation testing is done. Then user acceptance test is done to check whether the software satisfy its requirements.

CONCLUSION

6.0 CONCLUSION

The main aim of the project was to provide a good, fast and a reliable e-mail server, though not been fully updated, yet provides with the sophisticated use of e-mail system. This project has an effort at learning about .NET basics and networking.

Mails can be sent and received at whatever time of day suits the user. Security is provided for the user's mails and his information and also to the data available in the data base by backing up it. Multi domain support is provided for the users of various domains. The server is provided with the facility of logging and archiving the details with higher authority's permission. The instant messaging is an excellent feature to chat with the other users in the company.

The software has the deficiency of not sending audio and video files. It is normally difficult to teams for graphic files, spreadsheet and other non-text formats. The other systems must be loaded with the software for sending and receiving messages.

SCOPE FOR FURTHER DEVELOPMENT

7.0 SCOPE FOR FUTURE DEVELOPMENT

- The E-mail client side application used to read and compose mail will be developed instead of using the existing outlook.
- The anti spam facility will be made more advanced
- The anti virus feature will be developed
- All the features available will be made advanced

BIBLIOGRAPHY

8.0 BIBLIOGRAPHY

Books referred:

- Fred Barewell and Richard Blair, "Professional VB.NET", Wrox Publication, second edition, 2004.

- Steven Holzner, "Visual Basic .NET Programming Black Book", DreamTech Press, First edition,2003,Page No 1-57,109-247,588-605,833-927.

- E.Balagurusamy, "Object-Oriented Programming", Tata McGraw Hill Publication, 1999, Page No 1-25, 97-114.

- "MSDN (Microsoft Developers Network)", Microsoft Corporation, Visual Studio.NET Release.

Websites referred:

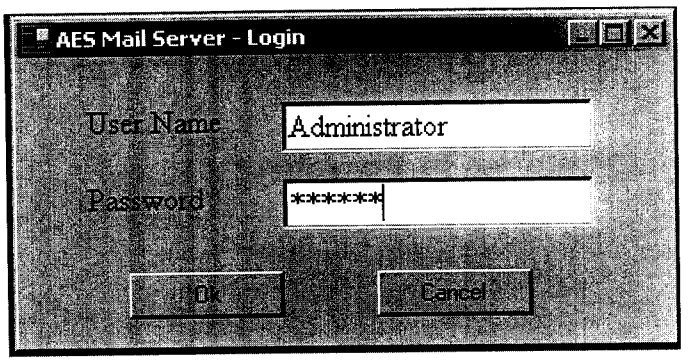
1. www.faqs.org
2. www.MerakMailServer.com
3. www.programmersheaven.com

APPENDIX

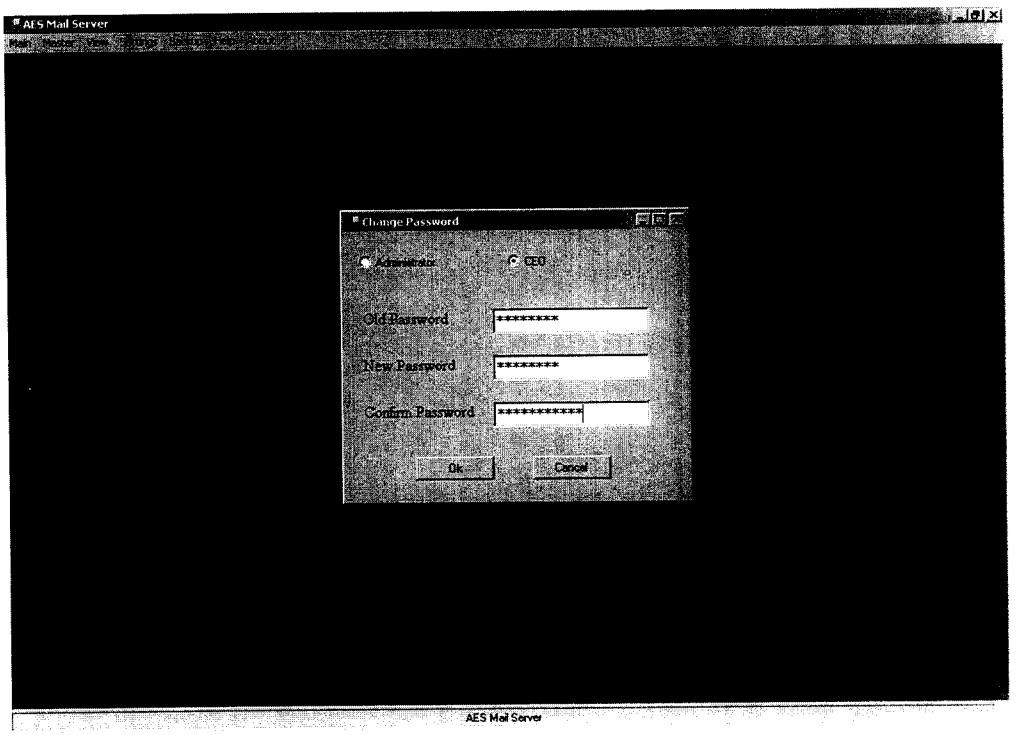
9.0 APPENDIX

9.1 SAMPLE SCREENS

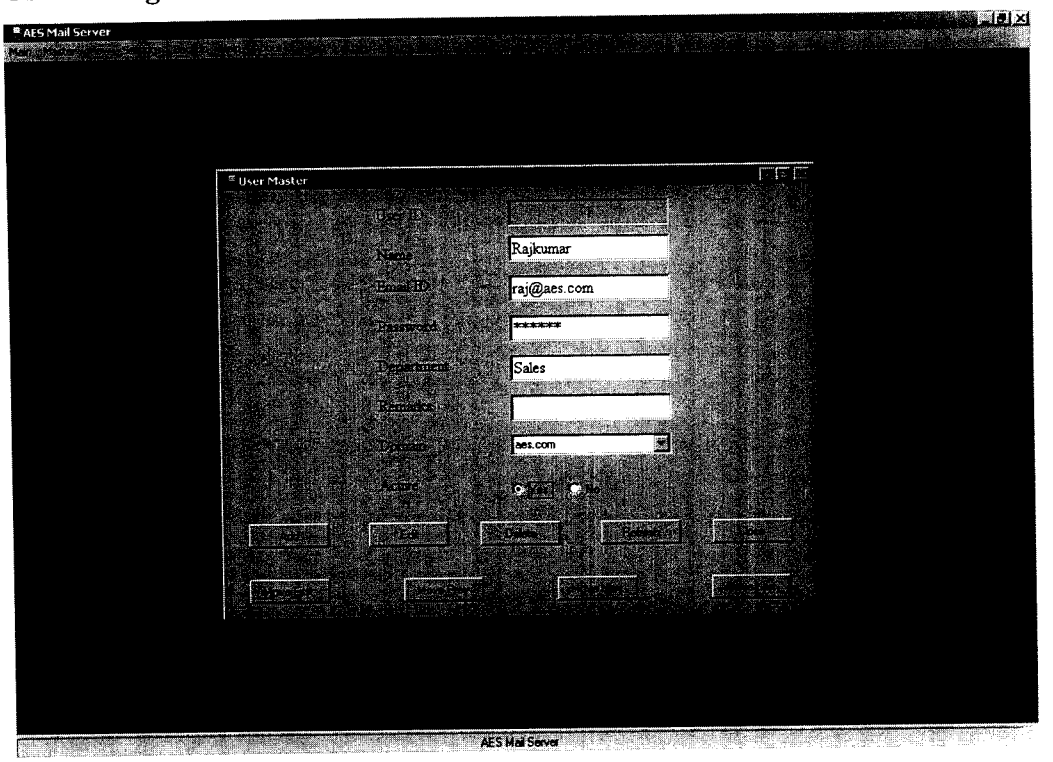
Login:



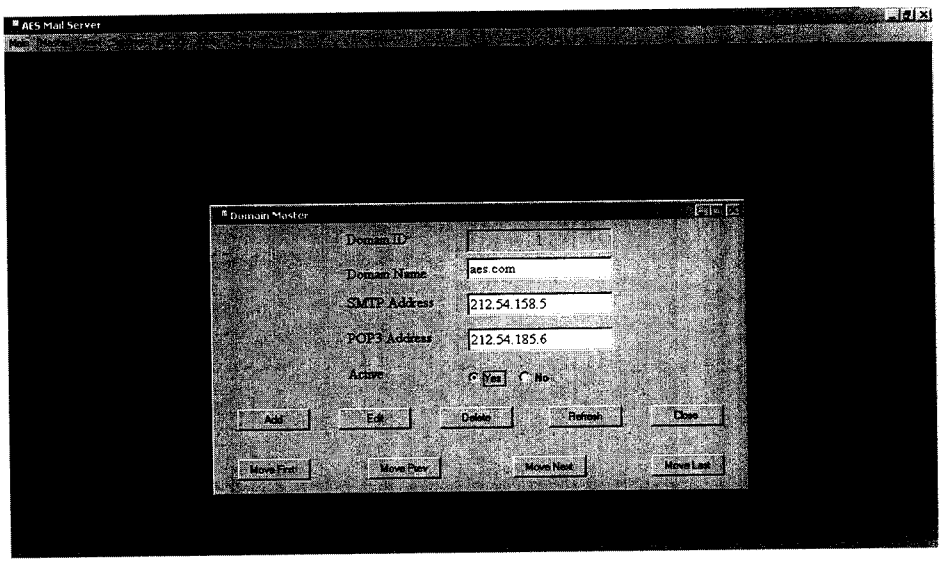
Change password:



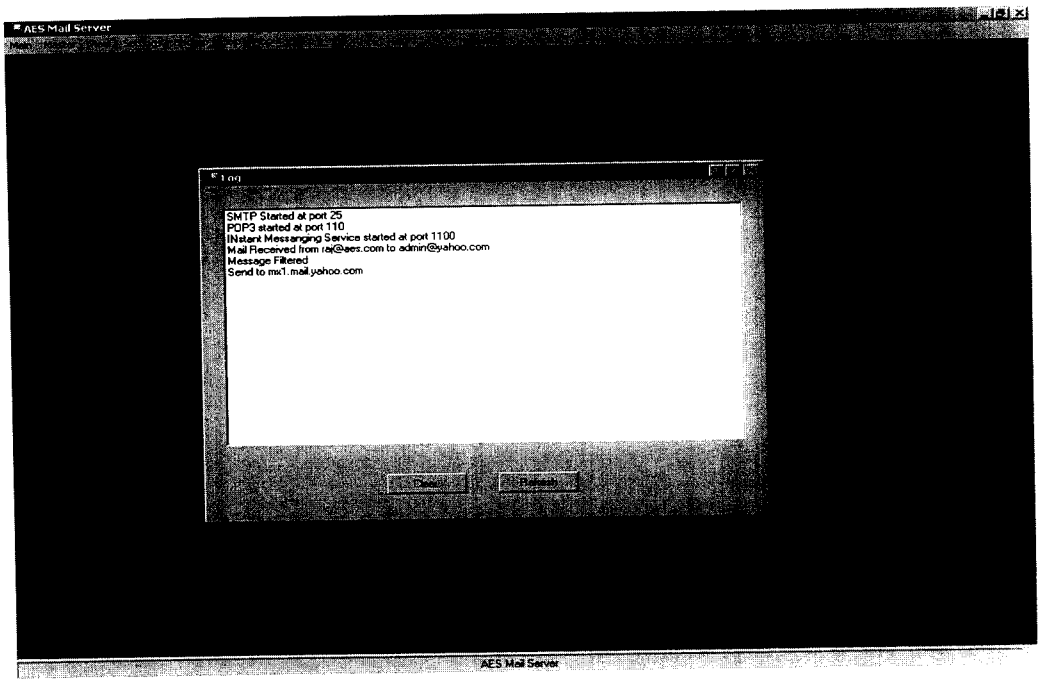
User settings:



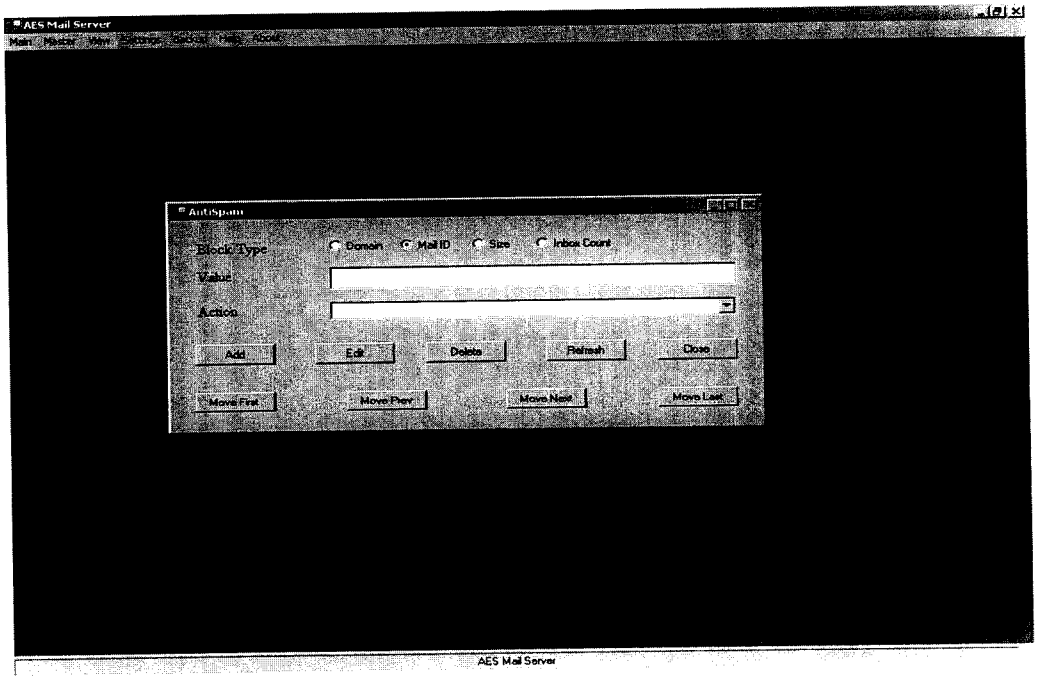
Domain settings :



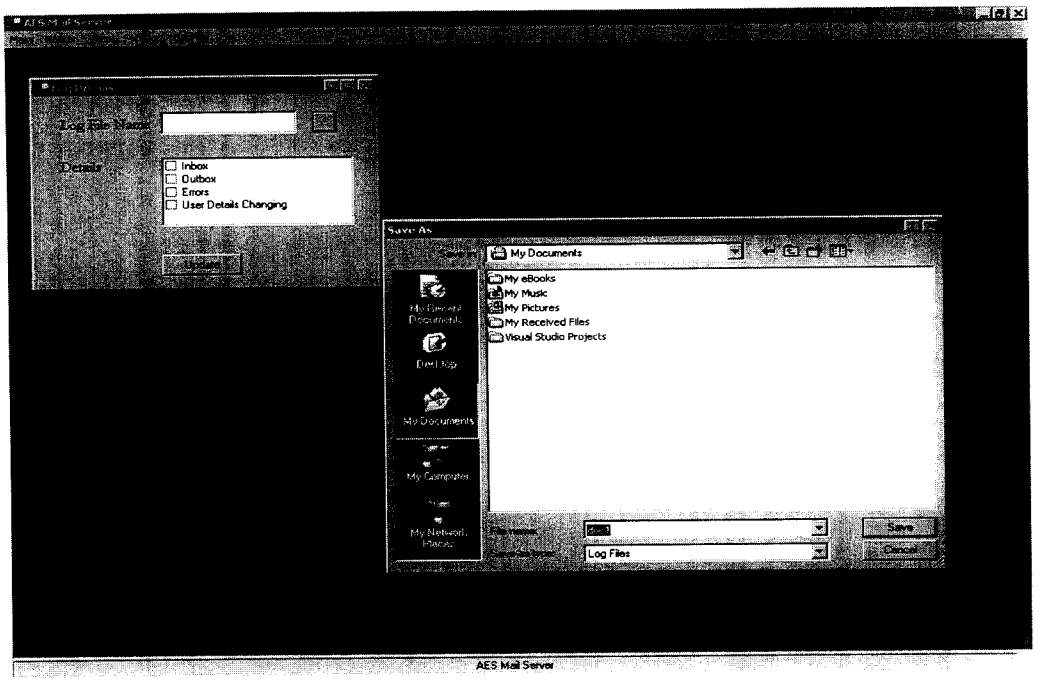
Logging:



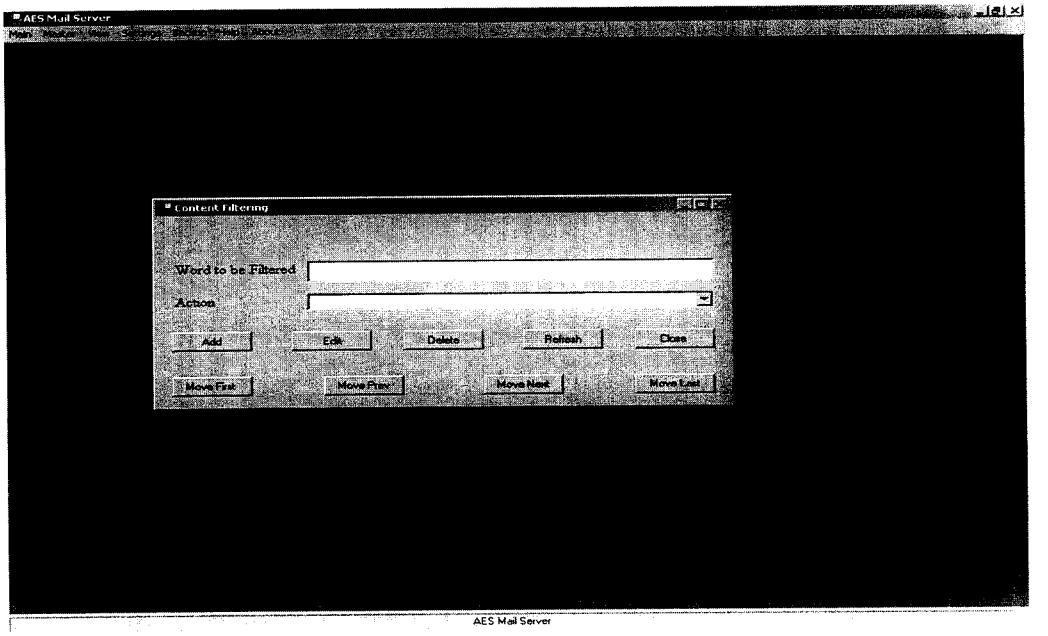
Anti spam:



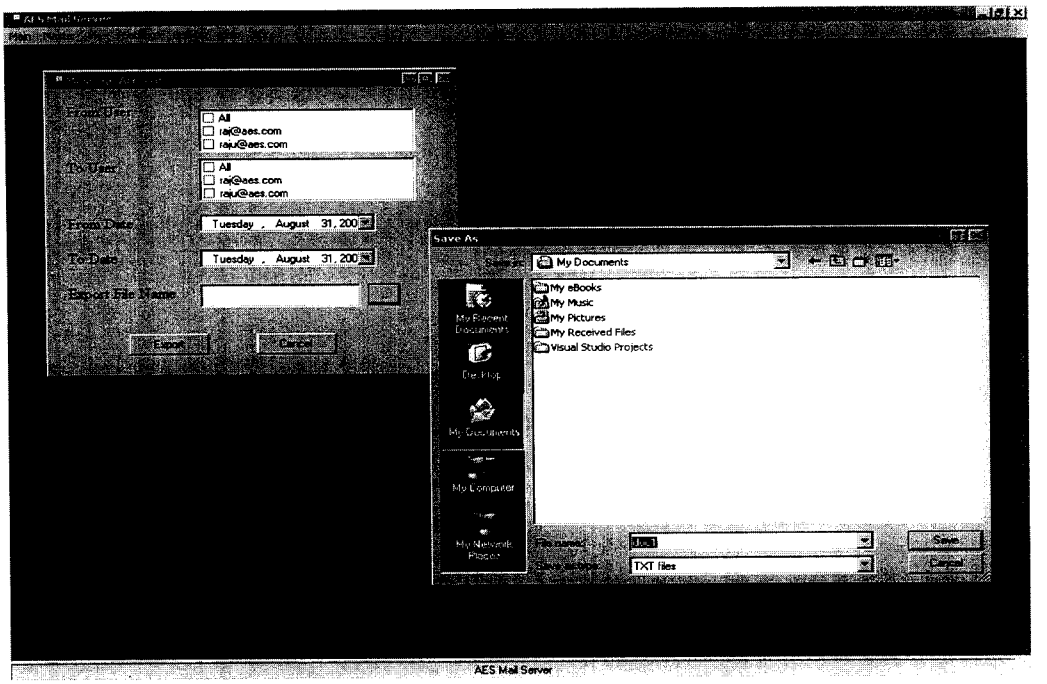
Log file:



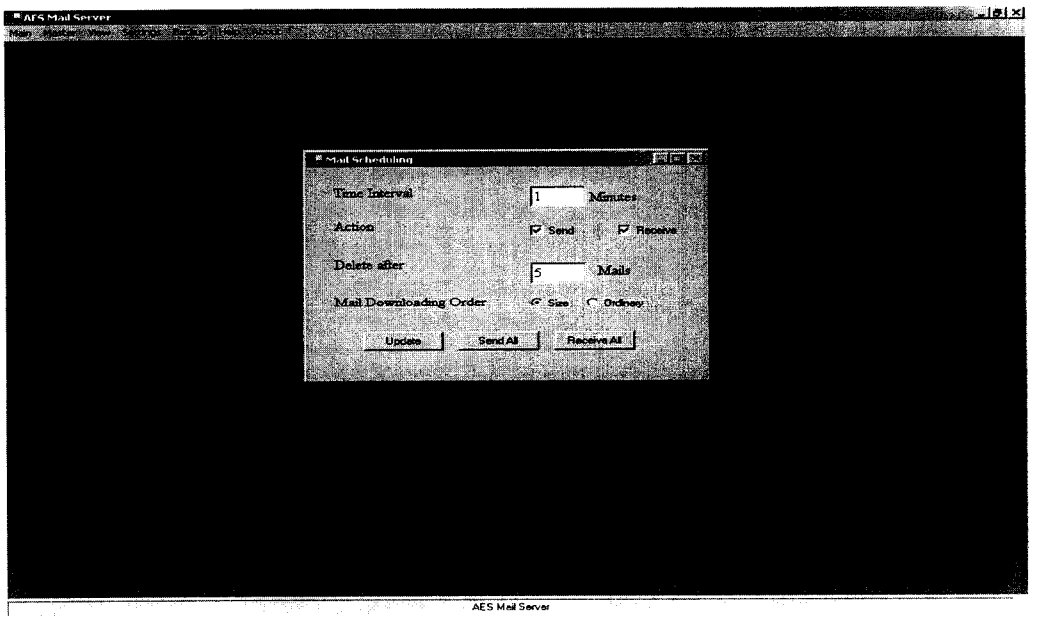
Content filtering:



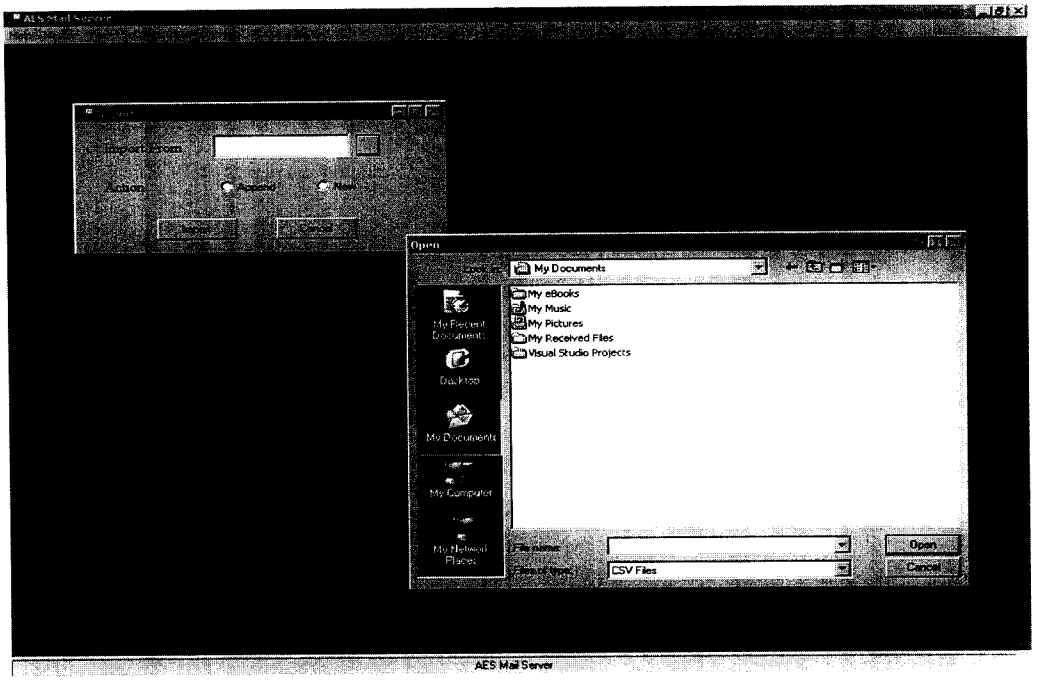
Archive filename:



Mail scheduling:



Back up Import:



Instant messaging public room: