

1. INTRODUCTION

1.1. ORGANIZATION PROFILE

2. ABOUT THE PROJECT

The Intranet mail server is communication software that helps the users of an Intranet to communicate with electronic mails. The mail service in the Internet is the most useful facility that enables the users of the Internet to send mails to any one who can be anywhere in the world. There is some free mail service providers such as HOTMAIL, YAHOO etc in the Internet. An user can have his account in any server like, praveen@hotmail.com, parthi@yahoo.com etc.

Now it is necessary to an organization to establish a communication path over its Intranet to facilitate electronic mail service. Also, the **Intranet Mail Server (IMS)** system provides the facility of mail service locally.

Since the system is a mail service. The system has to follow the Internet standards. In technical point of view, there are two modules that are the Client

module and the Server module. There are two protocols that have to be implemented for developing an mail software that are

1. POST OFFICE PROTOCOL 3 (POP 3) and
2. SIMPLE MAIL TRANSFER PROTOCOL (SMTP).

These protocols are nothing but set of commands. In IMS, both the server and the client used these protocols i.e. the commands that are defined in these protocols to make communication between themselves. The two protocols have to be implemented both in the server module and the client module.

The following figure shows the place where POP 3 and SMTP fit into the TCP/IP protocol suite.

The system is designed as Internet based application. The whole system is divided into following sub – system:

1. Server

User Agent

Mail Transfer Agent

2. Client

User Agent

Mail Transfer Agent

1.3. HARDWARE SPECIFICATION

PROCESSOR : Celeron 533 MHz

MEMORY : 64 MB RAM

HARD DISK : SEAGATE 10GB (6800 RPM)

MONITOR : 15” SAMTRON COLOR MONITOR.

KEYBOARD : Samsung Multimedia key board.

MOUSE : LOGITECH SCROLL MOUSE

FLOPPY DRIVE : 1.44 MB

NETWORK ADAPTOR PROPERLY CONFIGURED FOR TCP/IP

1.4. SOFTWARE SPECIFICATION

OPERATING SYSTEM : Windows 2000

LANGUAGE : JAVA

MAILING CLIENT : OUTLOOK EXPRESS

Features of Java

The inventors of Java wanted to design a language which could offer solutions to some of the problems encountered in modern programming. They wanted the language to be not only reliable, portable and distributed but also simple, compact and interactive. Sun Microsystems officially describes Java with the following attributes:

Compiled and Interpreted

Usually a computer language is either compiled or interpreted. Java combines both these approaches thus making Java a two-stage system. First, Java compiler translates source code into what is known as byte code instructions. Byte codes are not machine instructions and therefore, in the second stage, Java interpreter generates machine code that can be directly executed by the machine that is running the Java program. We can thus say that Java is both a compiled and an interpreted language.

Platform-Independent and Portable

The most significant contribution of Java over other languages is its portability. Java programs can be easily moved from one computer system to another, anywhere and anytime. Changes and upgrades in operating systems, processors and system resources will not force any changes in Java programs. This is the reason why Java has become a popular language for programming on Internet which interconnects different kinds of systems worldwide. We can download a Java applet from a remote computer onto our local system via Internet and execute it locally. This makes the Internet an extension of the user's basic system providing practically unlimited number of accessible applets and applications.

Java ensures portability in two ways. First, Java compiler generates byte code instructions that can be implemented on any machine. Secondly, the sizes of the primitive data types are machine-independent.

Object-Oriented

Java is a true object-oriented language. Almost everything in Java is an object. All program code and data reside within objects and classes. Java comes with an extensive set of classes, arranged in packages, that we can use in our programs by inheritance. The object model in Java is simple and easy to extend.

Robust and Secure

Java is a robust language. It provides many safeguards to ensure reliable code. It has strict compile time and run time checking for data types. It is designed as a garbage-collected language relieving the programmers virtually all memory management problems. Java also incorporates the concept of exception handling which captures series errors and eliminates any risk of crashing the system.

Security becomes an important issue for a language that is used for programming on Internet. Threat of viruses and abuse of resources is everywhere. Java systems not only verify all memory access but also ensure that no viruses are communicated with an applet. The absence of pointers in Java ensures that programs cannot gain access to memory locations without proper authorization.

Distributed

Java is designed as a distributed language for creating applications on networks. It has the ability to share both data and programs. Java applications can open and access remote objects on Internet as easily as they can do in a local

system. This enables multiple programmers at multiple remote locations to collaborate and work together on a single project.

Simple, Small and Familiar

Java is a small and simple language. Many features of C and C++ that are either redundant or sources of unreliable code are not part of Java. For example, Java does not use pointers, preprocessor header files, goto statement and many others. It also eliminates operator overloading and multiple inheritance.

Familiarity is another striking feature of Java. To make the language look familiar to the existing programmers, it was modeled on C and C++ languages. Java uses many constructs of C and C++ and therefore, Java code “looks like a C++” code. In fact, Java is a simplified version of C++.

Multithreaded and Interactive

Multithreaded means handling multiple tasks simultaneously. Java supports multithreaded programs. This means that we need not wait for the application to finish one task before beginning another. For example, we can listen to an audio clip while scrolling a page and at the same time download an applet from a distant computer. This feature greatly improves the interactive performance of graphical applications.

The Java runtime comes with tools that support multiprocess synchronization and construct smoothly running interactive systems.

High Performance

Java performance is impressive for an interpreted language, mainly due to the use of intermediate byte code. According to Sun, Java speed is comparable to the native C/C++. Java architecture is also designed to reduce overheads during runtime. Further, the incorporation of multithreading enhances the overall execution speed of Java programs.

Dynamic and Extensible

Java is a dynamic language. Java is capable of dynamically linking in new class libraries, methods, and objects. Java can also determine the type of class through a query, making it possible to either dynamically link or abort the program, depending on the response.

Java programs support functions written in other languages such as C and C++. These functions are known as native methods. This facility enables the programmers to use the efficient functions available in these languages. Native methods are linked dynamically at runtime.

2. SYSTEM ANALYSIS

2.1. PROBLEM DEFINITION

The initial state of the project is to either enter as an Administrator or already registered user. The Administrator has the rights to create or delete a particular user name, change the password and a separate inbox, compose, address book is provided for the Administrator. After creating id for the user by the administrator the user has the rights to enter into Intranet. The user alone is also provided with inbox, compose, address book, Trash option.

The main aim is to provide all the information that are needed by the user for his corresponding address.

2.2. SYSTEM STUDY

System survey is a general term that refers an orderly, structural process for identifying and solving problems. System analysis is a process, life cycle

methodology, since it relates to four significant phase in the life cycle of all business information systems: Study, Design, Development and Operation. The Definition of system analysis includes not only the process of analysis but also synthesis, which is the process of putting part together to form a new whole.

One of the major applications of Intranet communication is the mailing system. Mailing System enables one to send not only instant message but also to exchange files with anyone across the network. This project was developed using the language JAVA because it runs on any platform.

The other mailing system wholly relies on the mail server for its entire works to be done. Hence, the security and integration is problematic. Though mailing is less cost when compared to other communicative ways.

2.3. NEED FOR THE PROPOSED SYSTEM

This project handles all the jobs related to mails. The man-hours involved for transferring data can be reduced within an organization by using this mail facility. By entering the id and password the user or administrator can directly look into the information regarding mails. Hence, the security is maintained in the proposed system.

This project **INTRANET MAIL SERVER** is basically created because of the difficulties faced in the other mailing system. Hence, the mailing system is highly secured as compared to that of the other mailing systems.

To increase the portability and flexibility of the solution provided we have chosen JAVA. The server module is coded using JAVA.

OBJECTIVES OF PROPOSED SYSTEM

This project aims at developing an application that is capable of exchanging mails in an Intranet environment. Our project aims at developing the following:

- ❖ An application that identifies the user with the respective login-id and password.
- ❖ An inbox is maintained for each user to store incoming messages.
- ❖ An address book provision is given for each user to store address and mail ids.
- ❖ The Administrator has the facility to create and delete users.
- ❖ The users are also provided with the facility to reply and forward mails to any specified address.
- ❖ Users can send mail to many users and a clear identification of whether the mail has been sent or not sent is given.
- ❖ Users can also store their deleted messages in the trash.

ADVANTAGES FOR PROPOSED SYSTEM

INTRANET MAIL SERVER supports user friendliness. It reduces the communication gap between users whether it may be between higher authority and lower authority or between lower authorities or between lower authorities in an organization. This project is considered to be less cost effective. It is very feasible since its basic modules can be enhanced. JAVA supports the security system and enhances compatibility by providing exception-handling mechanisms.

3. SYSTEM DESIGN

The system design is the most creative and challenging phase. This is the process of developing the technical and operational specification of a system for implementation. Design is the first step in the development phase for any engineered products or system.

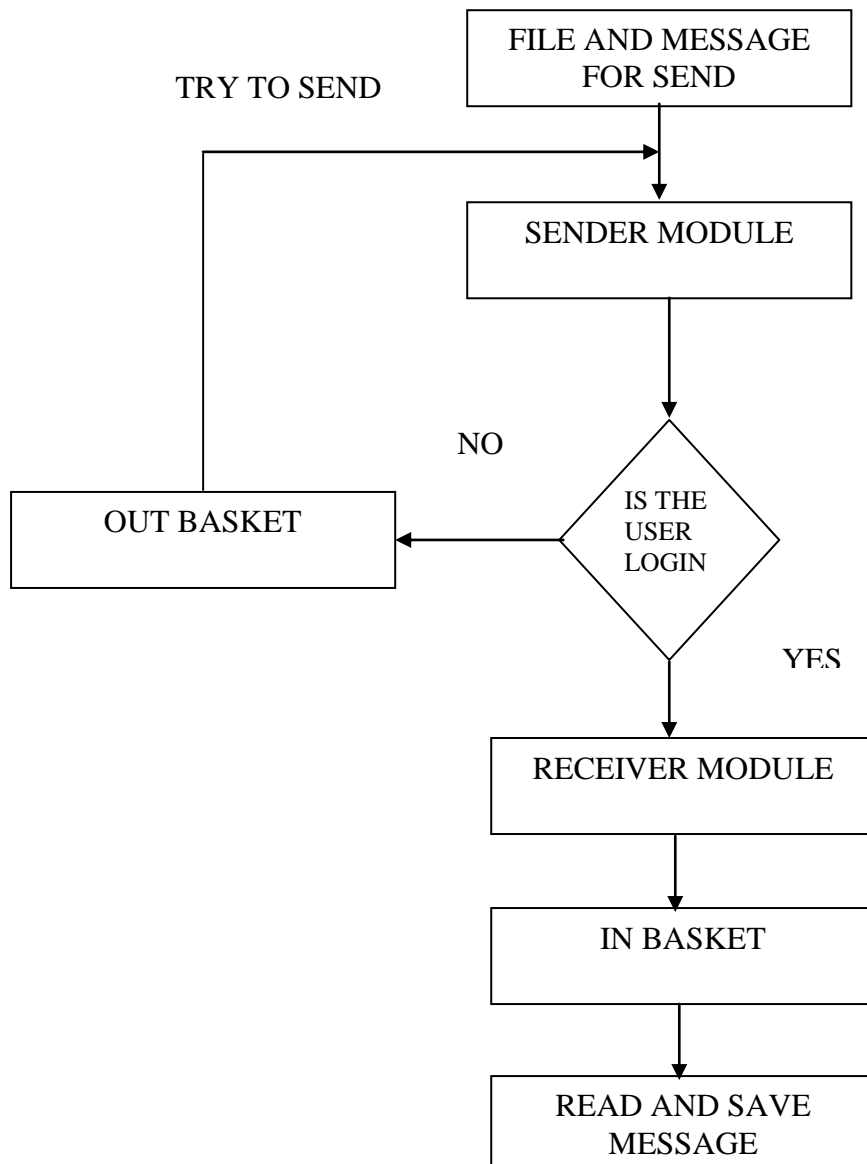
The design will determine the success of the system. Based on the proposed system objectives, the major modules are identified and the operations to be carried out are determined.

System design involves first logical design and then physical construction of the system. The first step in the design phase is to determine how the output produced and in what format and the second step consider about the input data.

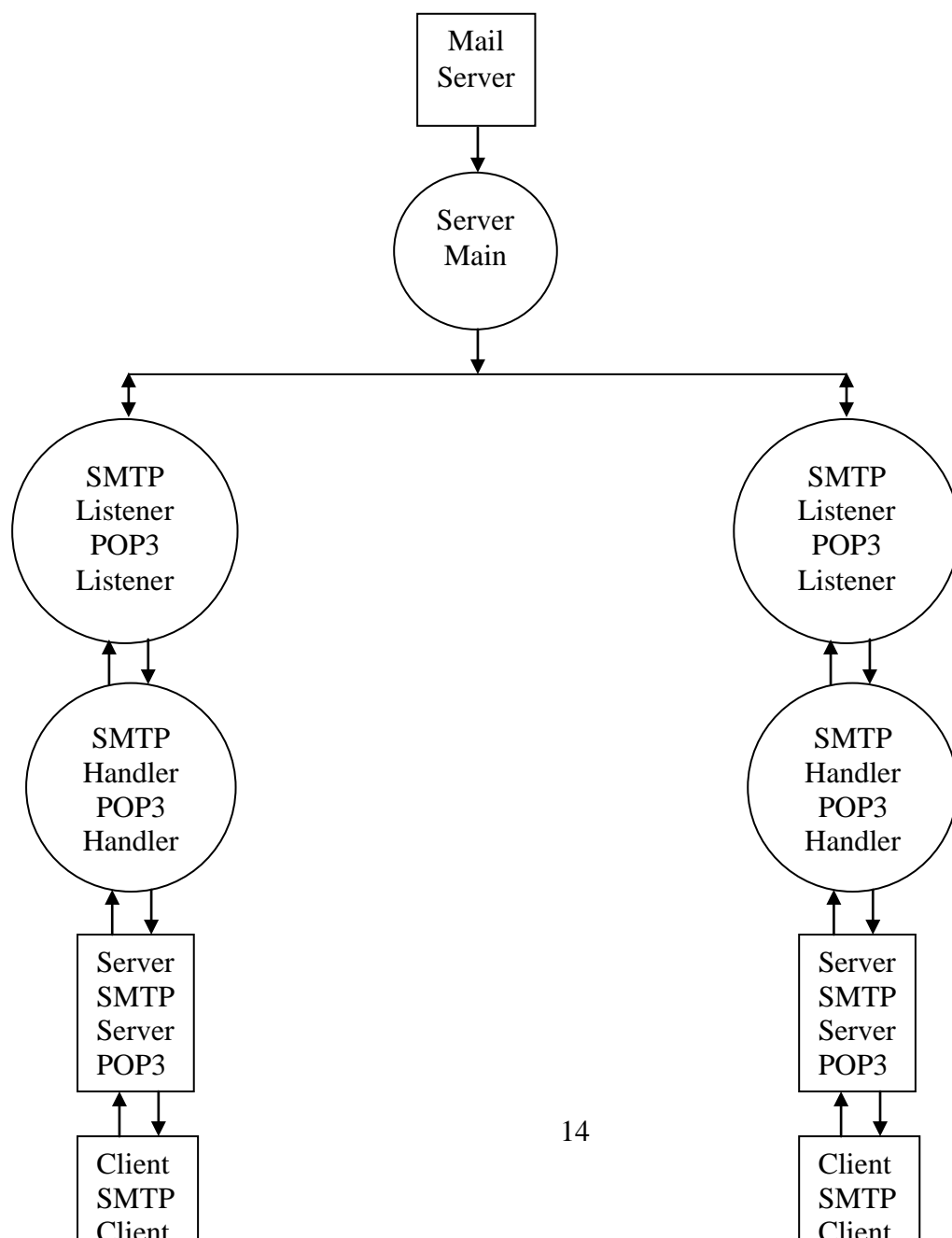
The operational phases are handled through program construction and testing. Software design resides at the technical kernel of the software engineering

process and is applied regardless of the development paradigm that is used. Software design is the first of three technical activities – design, code and test that are required to build and verify software.

3.1 SYSTEM FLOW



3.1.1 DATA FLOW DIAGRAM



3.2. MODULE SPECIFICATION

There are two modules in the intranet mail server. They are
Client module and
Server module

These modules are described as follows

CLIENT MODULE

Microsoft Outlook supports a variety of e-mail server types and communicates with these servers in different ways.

A MAPI-based architecture is provided if we use Outlook with Microsoft Exchange Server or with a MAPI-based store, such as a Personal Folders file (.pst). In general, the Outlook object model and developer-related features were designed for this environment. MAPI uses profiles to configure how an e-mail message is transmitted and where it is stored. Exchange server also provides many of the collaborative capabilities in Outlook, such as the Organizational Forms Library and public folders.

If Outlook is configured with an Internet Service Provider (ISP), typically using the POP protocol, you can create many types of Outlook solutions. However,

some of the topics in this documentation apply only when Outlook is used in conjunction with Microsoft Exchange Server, because that is the mode most commonly used to develop solutions based on Outlook.

The mailing client is Out Look Express. Some of the features of Out Look Express are:

Unique signature per account

Different default e-mail signature for each e-mail account.

New Data File Type (.pst)

Microsoft Office Outlook 2003 introduces a new file format for Personal Folders files (.pst) that offers greater storage capacity for items and folders and supports multilingual Unicode data. A file created with the new Microsoft Outlook Personal Folders file (.pst) format is not compatible with earlier versions of Outlook. For compatibility with earlier versions of Outlook, create a file by using the Microsoft Outlook 97-2002 Personal Folders file (.pst) format. Outlook Express can view and create files of either type.

Unicode Support

Outlook supports the Unicode text encoding standard, which allows Outlook to display text correctly, no matter what language it's written in, provided the operating system supports the characters specific to that language.

Integration with Microsoft Windows Share Point Services

View a Share Point event list next to your **Calendar** and drag events from the Share Point event list to your **Calendar**. We can also view Share Point contact lists from within Outlook and copy and link these to your **Contacts**. If we have access to a Microsoft Share Point Portal Server Web site, we can store a central calendar or contacts list on the server, for easy sharing of information.

POP enhancements

Download speed has been significantly increased, and we now get detailed progress information.

Tablet PC support

On a Tablet PC, you can quickly provide input using your own handwriting directly into Office documents as we would use a pen and a printout. Additionally, we can now view task panes horizontally to help we do our work on the Tablet PC the way we want to do our work.

Research task pane

The new **Research** task pane offers a wide variety of reference information and expanded resources if we have an Internet connection. We can conduct research on topics using an encyclopedia, Web search, or by accessing third-party content.

Microsoft Office Online

Microsoft Office Online is better integrated in all Microsoft Office programs so that you can take full advantage of what the site has to offer while we work. We can visit Microsoft Office Online directly from within our Web browser or use the links provided in various task panes and menus in our Office program to access articles, tips, clip art, templates, online training, downloads, and services to enhance how we work with Office programs. The site is updated regularly with new content based on direct feedback and specific requests from us and others who use Office.

Improving quality for the customer

Microsoft strives to improve quality, reliability, and performance of Microsoft software and services. The Customer Experience Improvement Program allows Microsoft to collect information about our hardware configuration and how we use Microsoft Office programs and services to identify trends and usage patterns. Participation is optional, and data collection is completely anonymous.

Additionally, error reporting and error messages have been improved so that we are provided with the easiest approach to reporting errors and the most helpful information about alerts at the time you encounter a problem. Finally, with an Internet connection, we can give Microsoft customer feedback about an Office program; help content, or Microsoft Office Online content. Microsoft is continually adding and improving content based on our feedback.

New in Intranet Mail Server

- ❖ **Global Address Book (GAB) module.** Centralized address book which can be accessed through a web browser from any machine in the network.
- ❖ Improved handling of various email clients.
- ❖ Behavior of "Retrieve new mails only" option. Earlier mails were marked as "read".

1. **SMTP Authentication support**
2. **Web interface for users.**

Web-based access to users' mailbox. - Intranet Mail Server users can access their mailbox from anywhere in the LAN.

Web interface for administrators to add, modify, delete accounts

3. Archive incoming and outgoing mails.

- ❖ **Automatic mail sending and receiving**

1. Automatic connects and disconnect facility for totally unmanned operation.
2. POP3 polling under automatic mode - Checks for mails at regular intervals.

- ❖ **Domain Names, Mailboxes and User options.**

1. Support for multiple mailboxes within your own domain name. eg: (anyusername@yourcompany.com) Typically when your ISP gives a POP3 mailbox which does the "catch all" for the mails received for anyuser@yourdomain.com Intranet Mail Server turns these individual mailboxes into virtual POP3 boxes with in the network.
2. Supports multiple POP3 accounts even from different domain names
3. Password protected mailboxes for each user.
4. Intranet email server - Routes mail to the same email domain locally, without the need to access your ISP
5. "Auto responder" facility for each user account to send a custom message automatically.
6. Email forwarding: Option to forward mails received by an user to another email address. This feature can be used for several purposes by the user. To mark a copy of messages received by an user to another user,
7. Rules Manager to filter messages based on keywords or phrases. we can also use this mail filtering facility to choose important/confidential messages to be delivered to your inbox or just trash those messages with abusive or junk content.

Other features:-

- ❖ Support for "POP before SMTP" authentication.
- ❖ Option to include/exclude a pop3 account profile during the receive operation.
- ❖ Can work behind a SOCKS 4 proxy.
- ❖ Based on Internet standards - SMTP / POP3
- ❖ Supports standard Email clients like Outlook Express, Eudora, Pegasus, Netscape Communicator and etc.
- ❖ Create mailboxes via easy to use interface.
- ❖ Turns any Windows 95 or NT system into an intranet mail gateway.
- ❖ Easy to install, configure and use; any novice can operate.

SERVER MODULE

Simple Mail Transfer Protocol (SMTP)

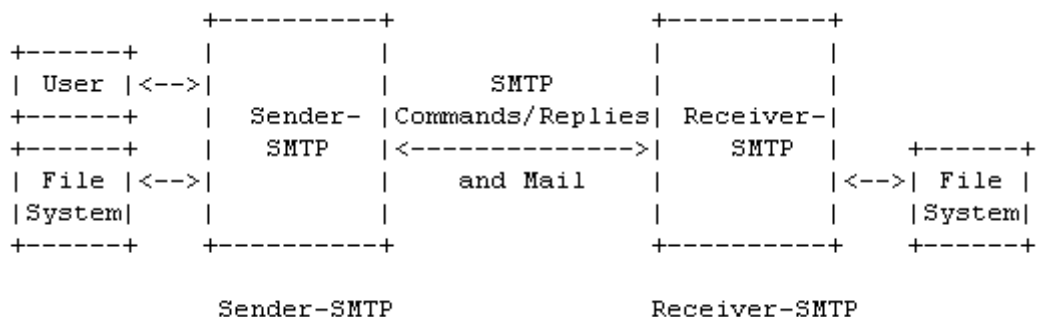
The objective of **Simple Mail Transfer Protocol (SMTP)** is to transfer mail reliably and efficiently. SMTP is independent of the particular transmission subsystem and requires only a reliable ordered data stream channel. An important feature of SMTP is its capability to relay mail across transport service environments. A transport service provides an Inter 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 interprocess 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 Model for SMTP use

The SMTP provides mechanisms for the transmission of mail; directly from the sending user's host to the receiving user's host when the two host are connected to the same transport service, or via one or more relay SMTP-servers when the source and destination hosts are not connected to the same transport service. To be able to provide the relay capability the SMTP-server must be supplied with the name of the ultimate destination host as well as the destination mailbox name.

The argument to the MAIL command is a reverse-path, which specifies who the mail is from. The argument to the RCPT command is a forward-path, which specifies who the mail is to. The forward-path is a source route, while the reverse-path is a return route (which may be used to return a message to the sender when an error occurs with a relayed message).

When the same message is sent to multiple recipients the SMTP encourages the transmission of only one copy of the data for all the recipients at the same destination host.

The mail commands and replies have a rigid syntax. Replies also have a numeric code. In the following, examples appear which use actual commands and replies. The complete lists of commands and replies appears in Section 4 on specifications.

Commands and replies are not case sensitive. That is, a command or reply word may be upper case, lower case, or any mixture of upper and lower case. Note that this is not true of mailbox user names. For some hosts the user name is case sensitive, and SMTP implementations must take case to preserve the case of user names as they appear in mailbox arguments. Host names are not case sensitive. Commands and replies are composed of characters from the ASCII character set [1]. When the transport service provides an 8-bit byte (octet) transmission channel, each 7-bit character is transmitted right justified in an octet with the high order bit cleared to zero.

The SMTP Procedures

This section presents the procedures used in SMTP in several parts. First comes the basic mail procedure defined as a mail transaction. Following this are descriptions of forwarding mail, verifying mailbox names and expanding mailing

lists, sending to terminals instead of or in combination with mailboxes, and the opening and closing exchanges.

MAIL

There are three steps to SMTP mail transactions. The transaction is started with a MAIL command which gives the sender identification. A series of one or more RCPT commands follows giving the receiver information. Then a DATA command gives the mail data. And finally, the end of mail data indicator confirms the transaction. The first step in the procedure is the MAIL command. The <reverse-path> contains the source mailbox.

MAIL <SP> FROM:<reverse-path> <CRLF>

This command tells the SMTP-receiver that a new mail transaction is starting and to reset all its state tables and buffers, including any recipients or mail data. It gives the reverse-path which can be used to report errors. If accepted, the receiver-SMTP returns OK reply.

The <reverse-path> can contain more than just a mailbox. The <reverse-path> is a reverse source routing list of hosts and source mailbox. The first host in the <reverse-path> should be the host sending this command. The second step in the procedure is the RCPT command.

RCPT <SP> TO :<forward-path> <CRLF>

This command gives a forward-path identifying one recipient. If accepted, the receiver-SMTP returns OK reply, and stores the forward-path. If the recipient is unknown the receiver-SMTP returns Failure reply. This second step of the procedure can be repeated any number of times. The <forward-path> can contain more than just a mailbox. The <forward-path> is a source routing list of hosts and

the destination mailbox. The first host in the <forward-path> should be the host receiving this command. The third step in the procedure is the DATA command.

DATA <CRLF>

If accepted, the receiver-SMTP returns Intermediate reply and considers all succeeding lines to be the message text. When the end of text is received and stored the SMTP-receiver sends OK reply.

Since the mail data is sent on the transmission channel the end of the mail data must be indicated so that the command and reply dialog can be resumed. SMTP indicates the end of the mail data by sending a line containing only a period. A transparency procedure is used to prevent this.

The mail data includes the memo header items such as Date, Subject, To, Cc and from. The end of mail data indicator also confirms the mail transaction and tells the receiver-SMTP to now process the stored recipients and mail data. If accepted, the receiver-SMTP returns OK reply. The DATA command should fail only if the mail transaction was incomplete (for example, no recipients), or if resources are not available.

Post Office Protocol 3 (POP3)

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 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 post 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. Responses to certain commands are multi-line. In these cases, which are clearly indicated below, after sending the first line of the response and a CRLF, any additional lines are sent, each terminated by a CRLF pair. When all lines of the response have been sent, a final line is sent, consisting of a termination octet (decimal code 046, ".") and a CRLF pair. If any line of the multi-line response

begins with the termination octet, the line is "byte-stuffed" by pre-pending the termination octet to that line of the response.

Hence a multi-line response is terminated with the five octets "CRLF.CRLF". When examining a multi-line response, the client checks to see if the line begins with the termination octet. If so and if octets other than CRLF follow, the first octet of the line (the termination octet) is stripped away. If so and if CRLF immediately follows the termination character, then the response from the POP server is ended and the line containing ".CRLF" is not considered part of the multi-line response.

A POP3 session progresses through a number of states during its lifetime. Once the TCP connection has been opened and the POP3 server has sent the greeting, the session enters the AUTHORIZATION state. In this state, the client must identify itself to the POP3 server. Once the client has successfully done this, the server acquires resources associated with the client's mail drop, and the session enters the TRANSACTION state. In this state, the client requests actions on the part of the POP3 server. When the client has issued the QUIT command, the session enters the UPDATE state. In this state, the POP3 server releases any resources acquired during the TRANSACTION state and says goodbye. The TCP connection is then closed. A server MUST respond to an unrecognized, unimplemented, or syntactically invalid command by responding with a negative status indicator. A server MUST respond to a command issued when the session is in an incorrect state by responding with a negative status indicator. There is no general method for a client to distinguish between a server which does not implement an optional command and a server which is unwilling or unable to process the command.

A POP3 server MAY have an inactivity auto logout timer. Such a timer MUST be of at least 10 minutes' duration. The receipt of any command from the client during that interval should suffice to reset the auto logout timer. When the timer expires, the session does NOT enter the UPDATE state--the server should

close the TCP connection without removing any messages or sending any response to the client.

4. SYSTEM TESTING AND IMPLEMENTATION

TESTING

Testing is the set of activities that can be planned in advance and conducted systematically. It is a process of executing programs with the intention of finding errors.

SYSTEM TESTING

Software testing is a critical element of software quality assurance and represents the ultimate view of specification, design and coding. The user tests the developed system and changes are made according to their needs. The testing phase involves the testing of developed system using various kinds of data.

System testing is an important task in implementing the system. The system tested with the best case values and worst case values for its response and

not produce an undesirable runtime errors and providing the user with proper message and correcting the wrong inputs.

System testing is a style of implementation which is aimed at ensuring works at all levels and effective before live operation starts. The system test before implementation should give confirmation that all is correct and it provides opportunity to show the users that the system works well.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently before the operations commence. Testing is vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests. A series of testing is performed for the proposed system before the system is ready for the acceptance test.

The following tests are conducted

1. Module Testing
2. Integration Testing
3. User Acceptance Testing

Module Testing

Unit testing focuses verification effort on the smallest unit of the software design module. This is known as module testing. The modules of the Job Consultancy Management System are tested separately. This testing was carried out during programming stage itself. In this testing step each module was found to be working satisfactorily with regard to the expected output from the module.

Integration Testing

Data can be lost across an interface; one module can have an adverse effect on another, sub functions when combined, may not produce the desired major function. Integration Testing is a systematic technique for constructing errors associated with in the interface. The objective is to take unit tested modules and to build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the isolation of causes is complicated by the vast expenses of the entire program. Thus in the integration testing step, all the errors uncovered are corrected for the next testing steps.

User Acceptance Testing

User acceptance is a key factor for a success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the perspective system users at the time of developing and making changes whenever required. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using the test data. While testing the system using the test data errors are again uncovered. These

errors are again corrected by using the above testing steps and corrections are also noted for future use.

IMPLEMENTATION

A crucial phase in the system life cycle is the successful implementation of the new system.

Implementation simply means converting a new system design into operation.

There are basically three types of implementation such as

1. Implementation of computer system to the replacement of manual system.
2. Implementation of new system to replace existing one.
3. Implementation of a modified application to replacement of an existing one.

The implementation of **INTRANET MAIL SERVER** comes under the first category and this project is implemented in

5. CONCLUSION AND SUGGESTIONS

CONCLUSION

This project “INTRANET MAIL SERVER” on so far has been described tested and documented completely. Hence, the same project could be used so far INTRANET and INTERNET.

To increase the portability and flexibility of the solution provided we have chosen JAVA. The server module is coded using JAVA Servlets. This software is attractive and it allows easy way to use mail for the user. The online time project is to be maintained by the company.

SUGGESTIONS

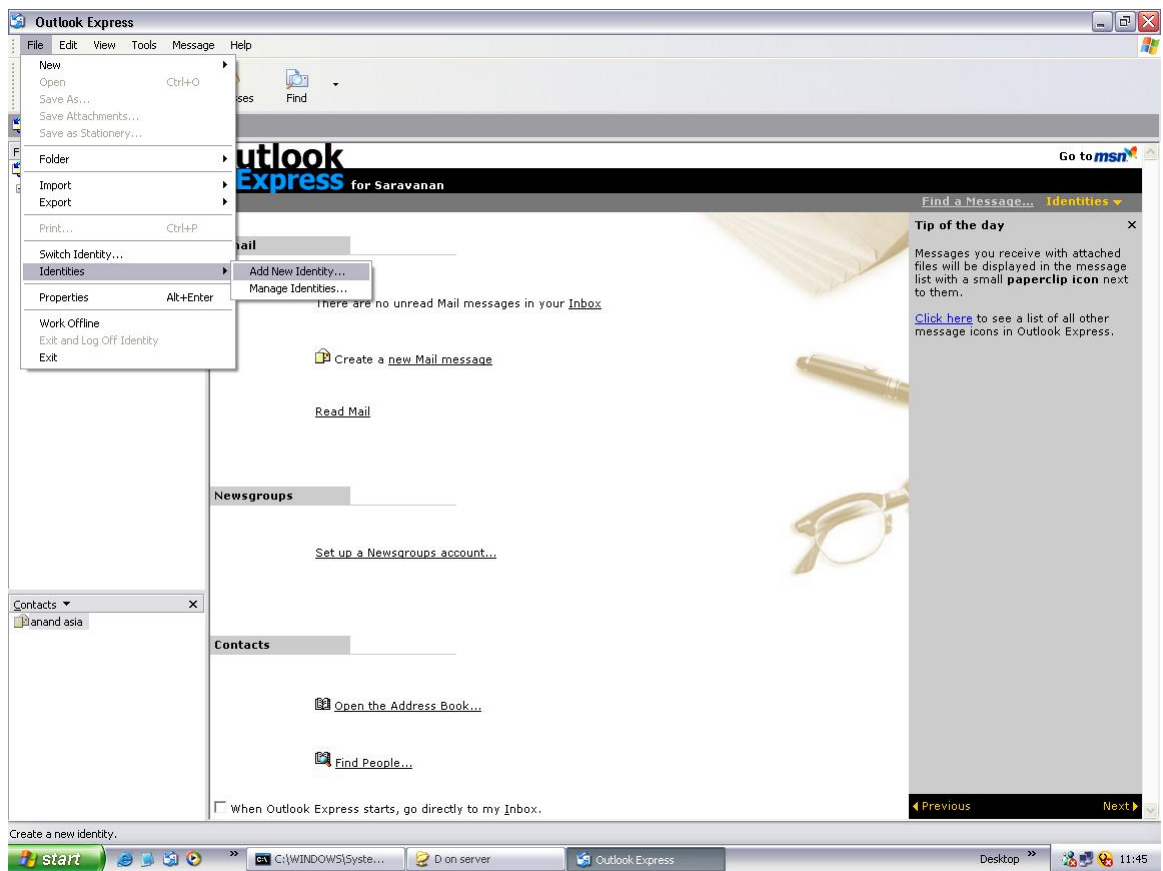
Including certain needs can extend this project. Amidst a lot of advantages, including the following features can enhance the proposed system.

By providing portability the INTRANET MAIL SERVER holds an Outlook Express.

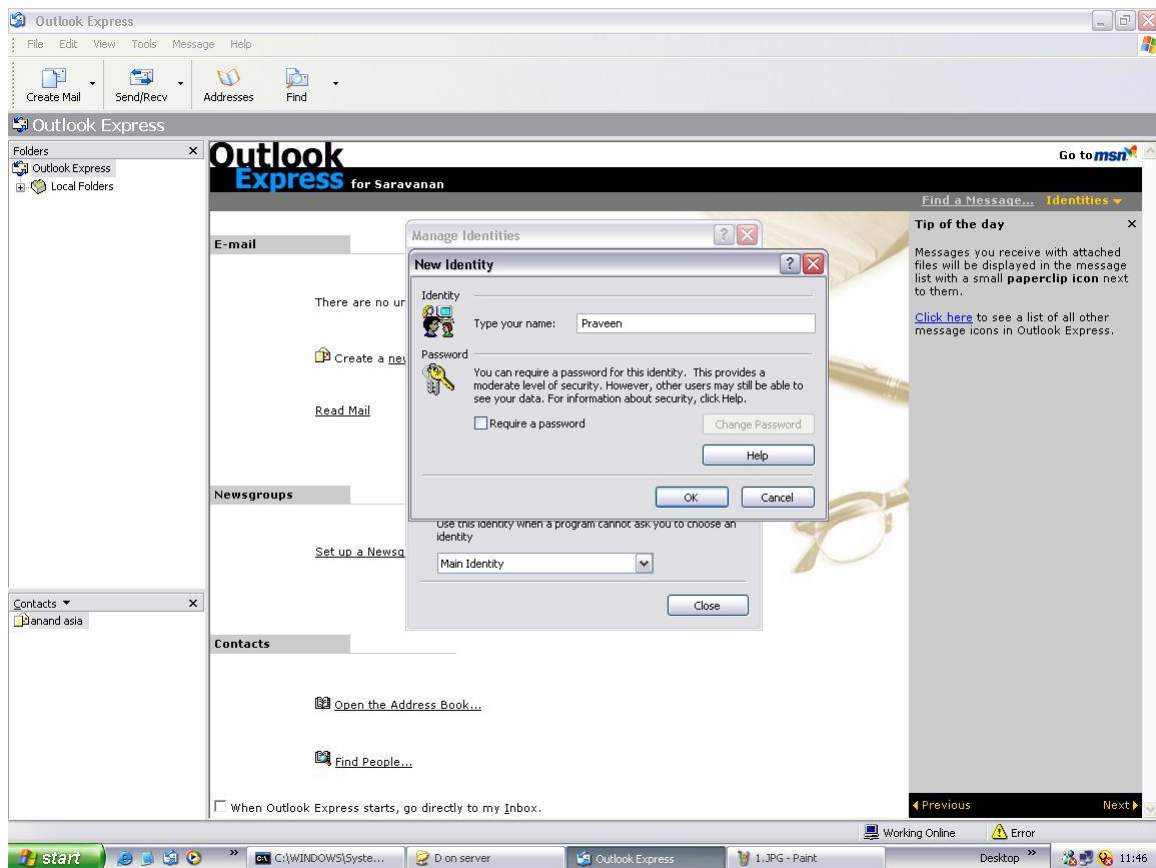
Creation, deletion and updating of folders can be done. Also, an e group can also be formed within the INTRANET MAIL.

BIBLIOGRAPHY

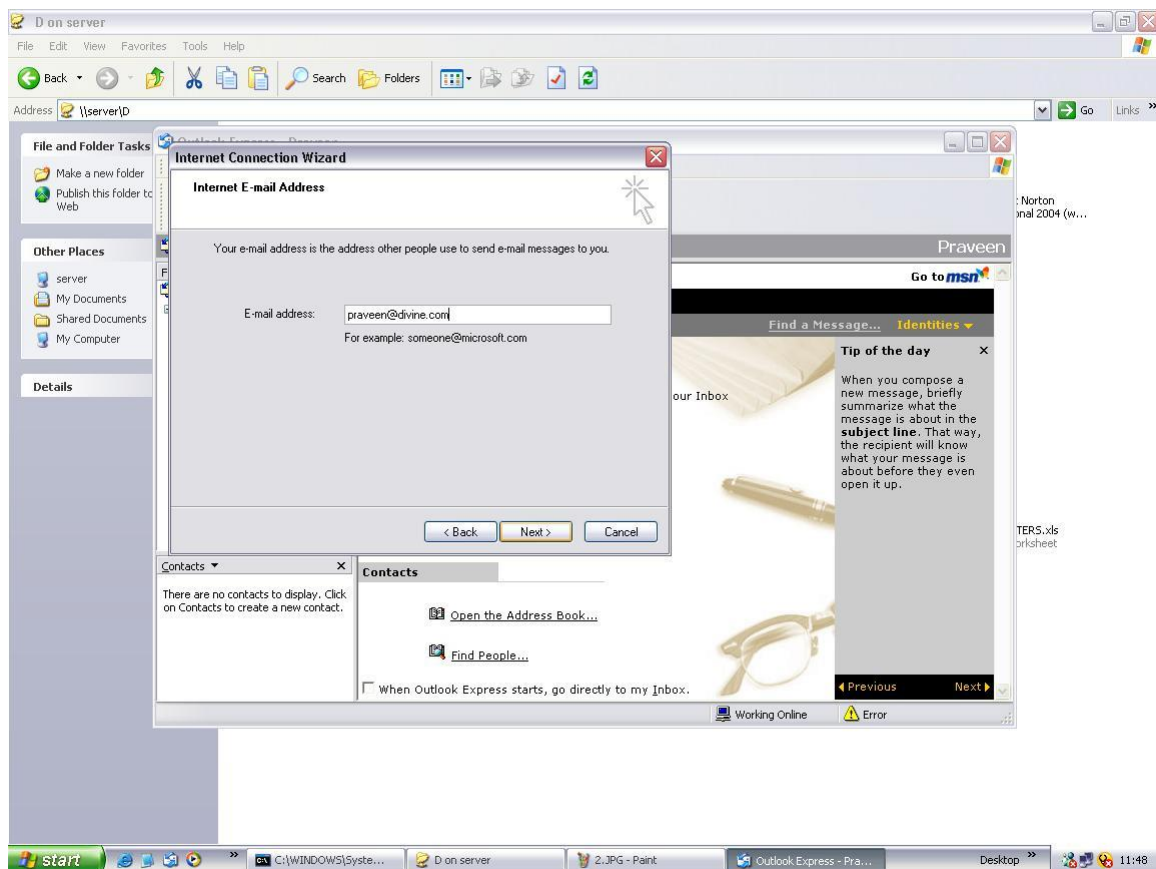
1. **Elias M.Awad**, “*System Analysis and Design*”, Galgotia Publications, Second Edition,1998.
2. **Lan Sommerville**,”*Software Engineering*”, Pearson Publications”, Sixth Edition,2001.
3. **Patric Naughton and Herbert Schildt**, “ *Complete Reference (JAVA 2)*”, Tata McGRAW Hill Publications Company Ltd.,Fifth Edition,2002.



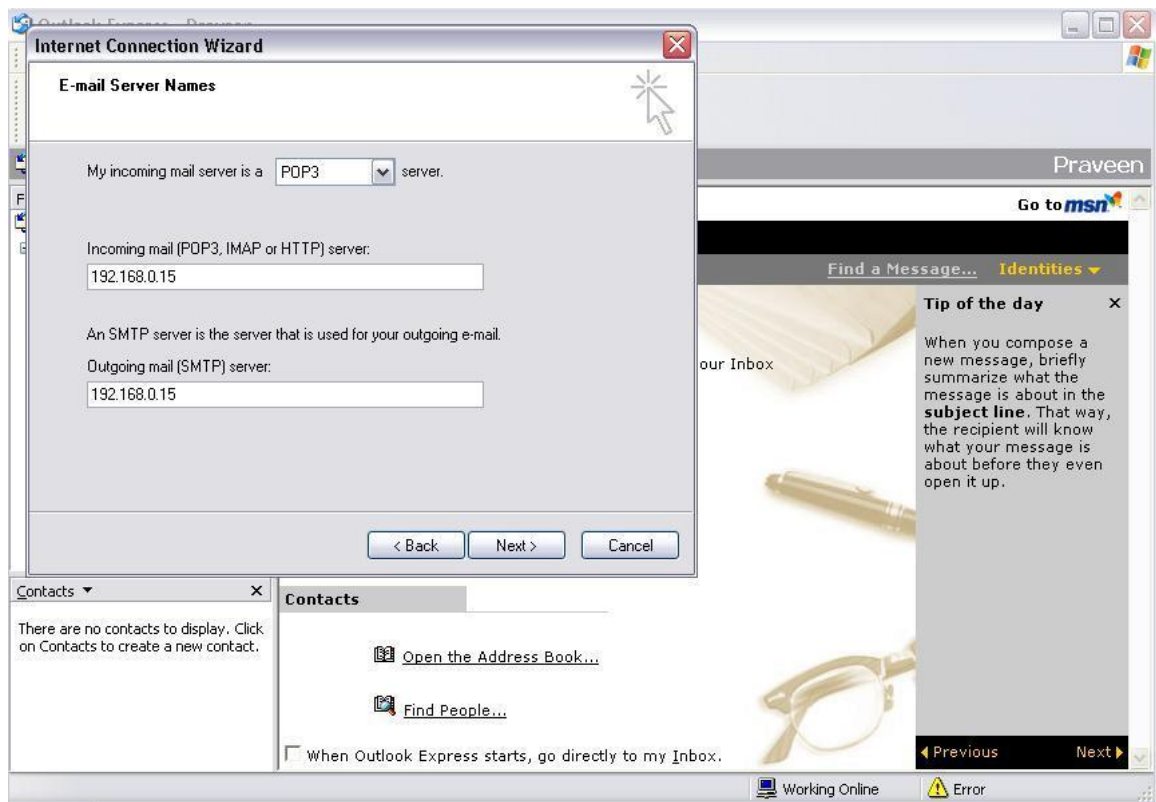
A-1 CREATING AN IDENTITY



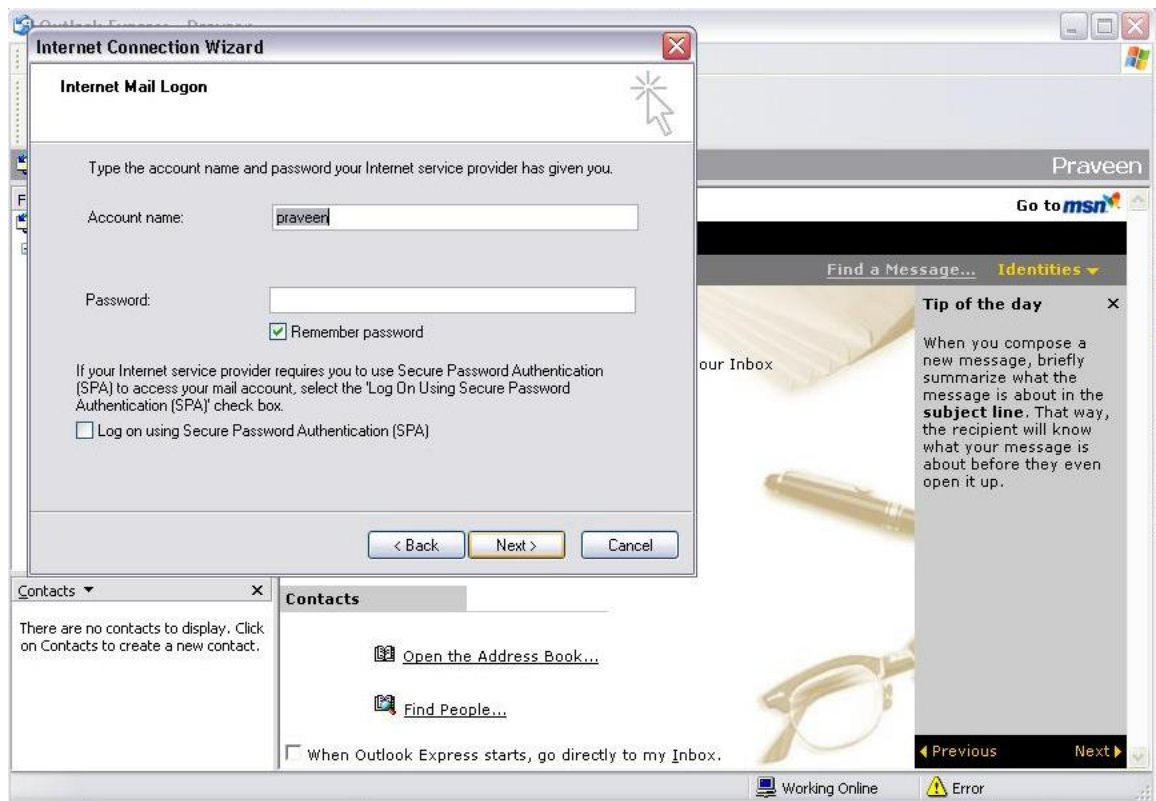
A-2 GIVING THE DETAILS OF THE CLIENT



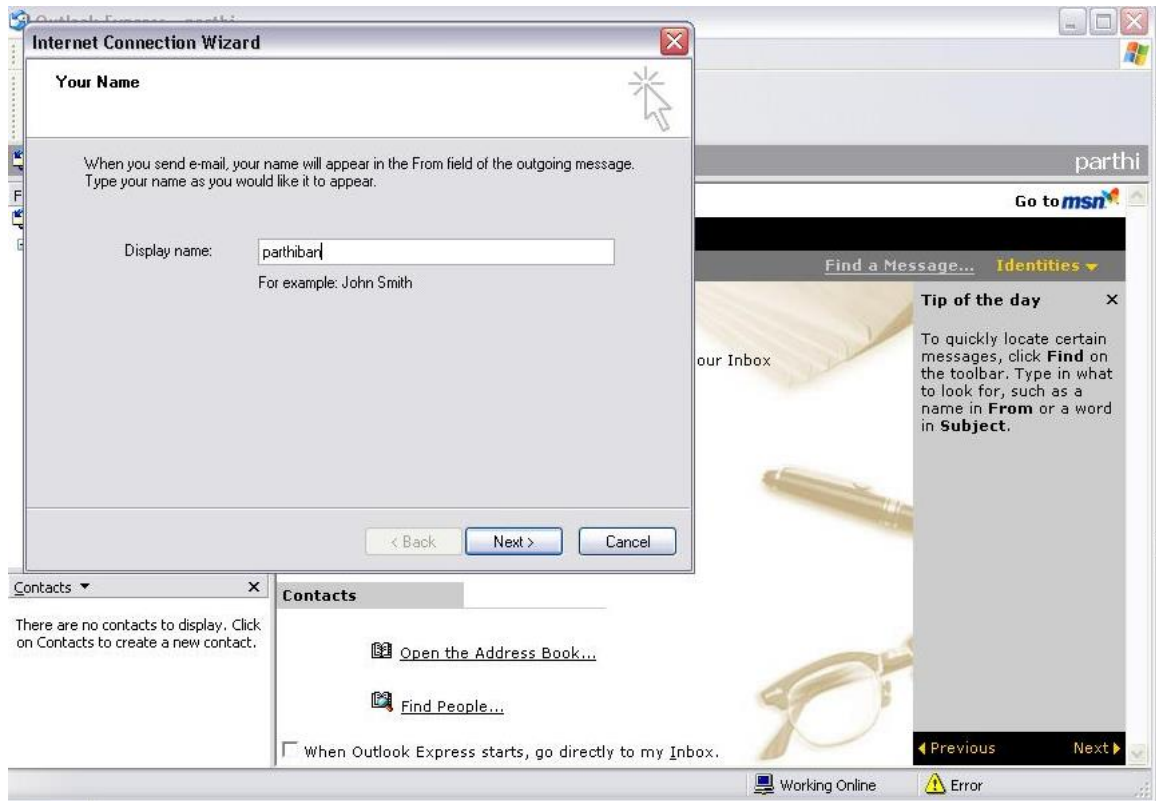
A-3 CREATING THE MAILING ADDRESS



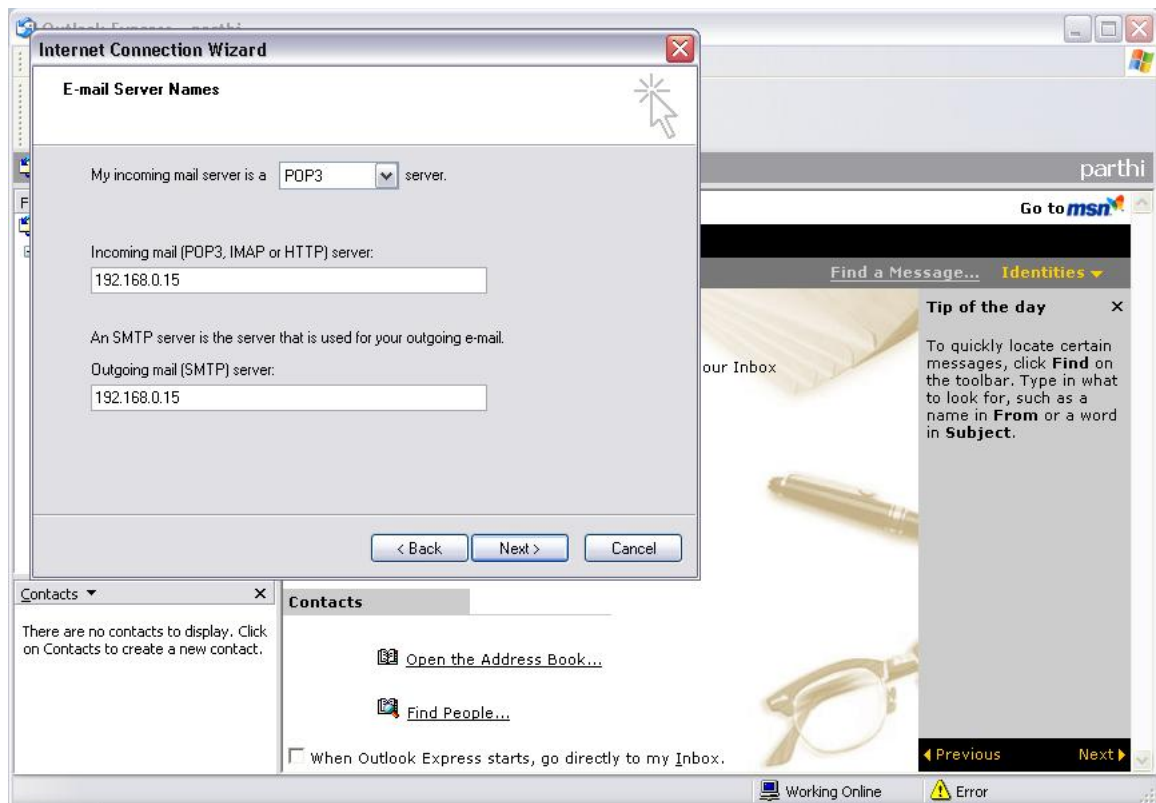
A-4 SETTING THE IP ADDRESS TO THE CLIENT



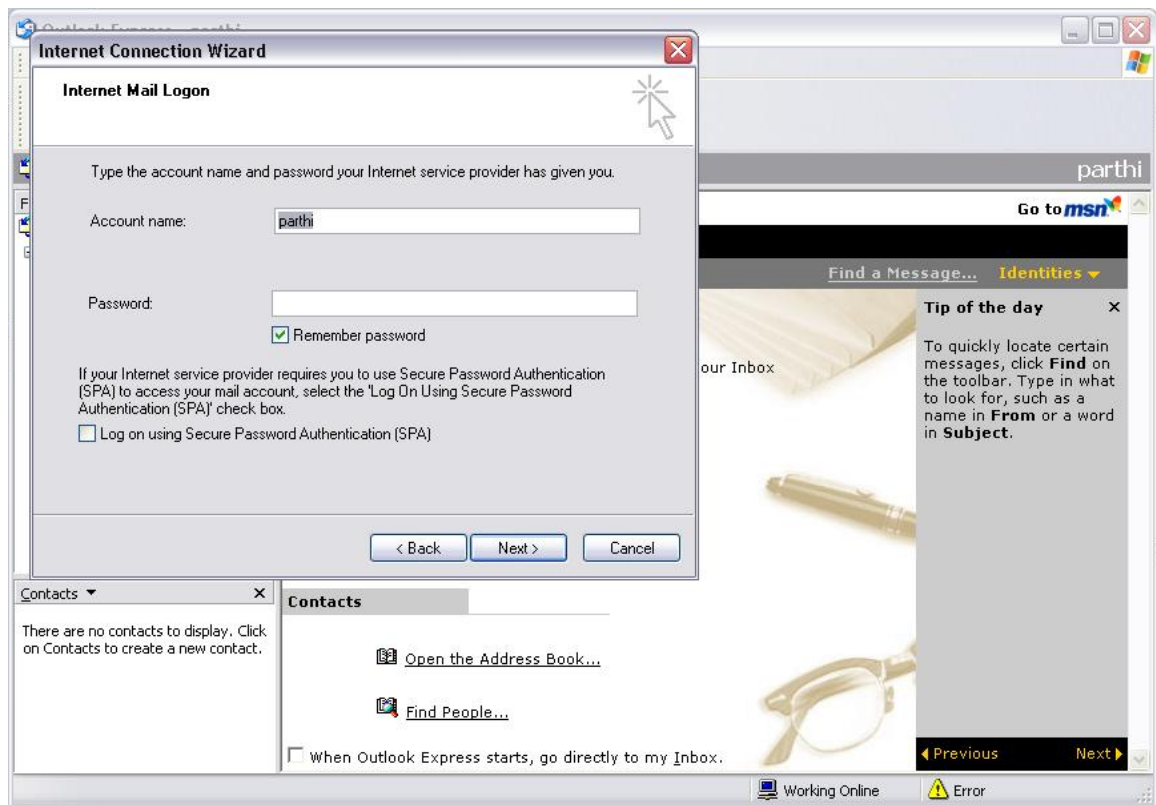
A-5 SETTING THE PASSWORD TO THE CLIENT



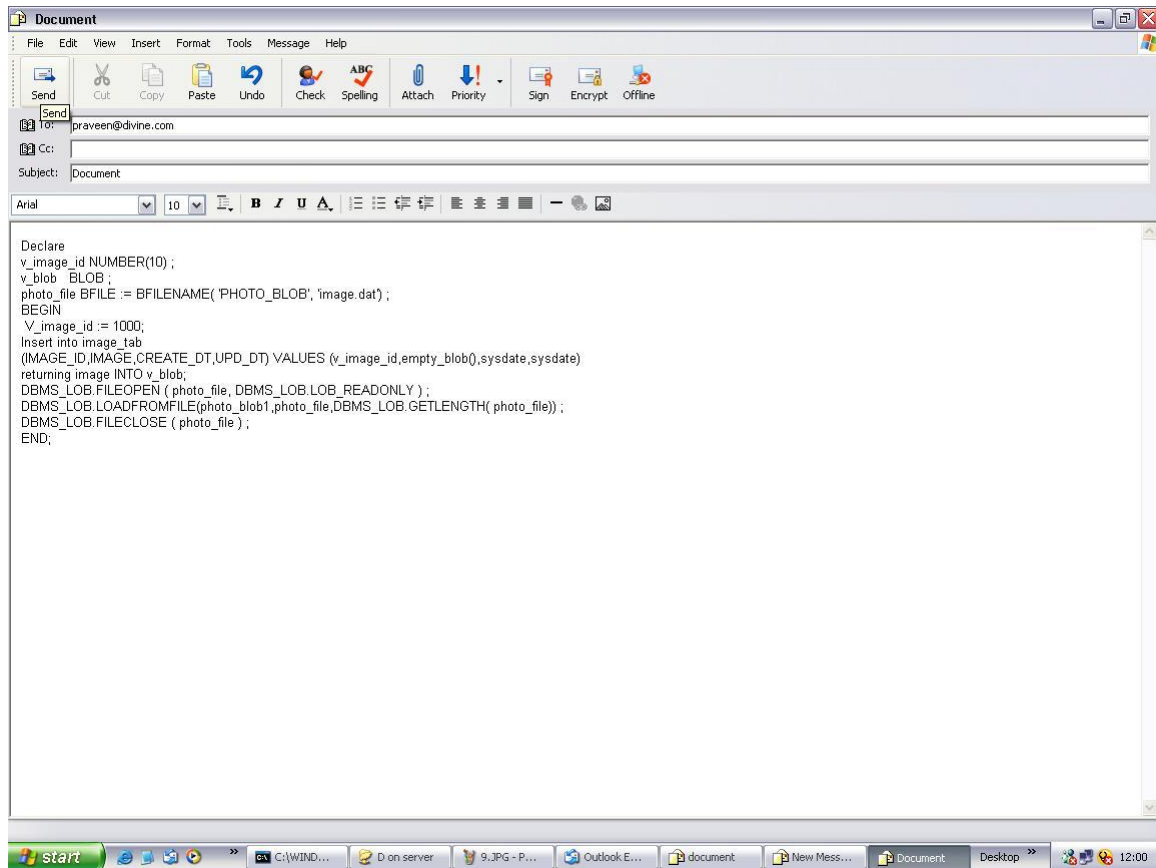
A-6 GIVING THE DETAILS OF THE CLIENT



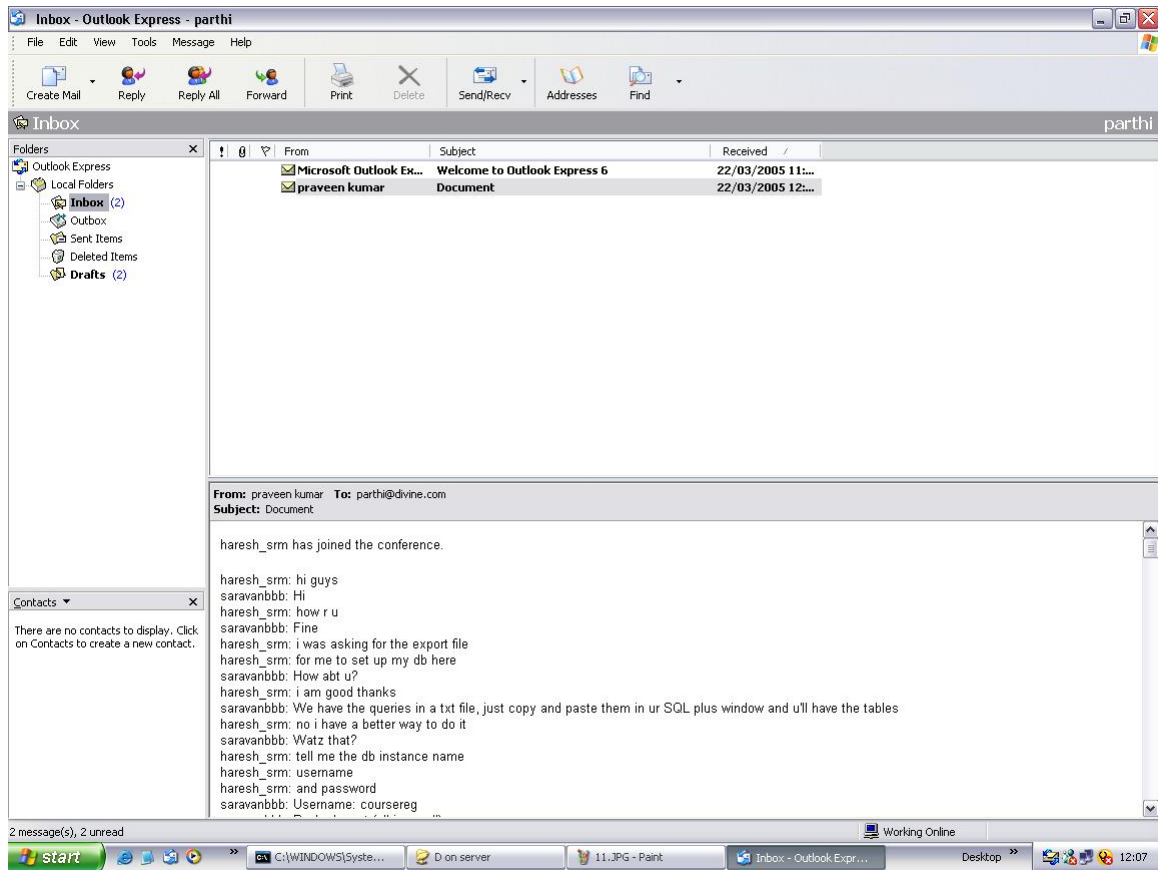
A-7 SETTING THE IP ADDRESS TO THE CLIENT



A-8 SETTING THE PASSWORD TO THE CLIENT



A-9 SENDING THE MESSAGE



A-10 RECEIVING THE MESSAGE



A screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\System32\cmd.exe - java ServerMain 27 110". The command prompt shows the following text:

```
C:\j2sdk1.4.1_04\bin\JM>java ServerMain 27 110
SMTP (send) port = 27
POP3 (receive) port = 110
Starting POP3 listener
Completed Java Mail Server SMTP
Starting SMTP listener
```

B-1 SERVER FORM

```
C:\WINDOWS\System32\cmd.exe - java ServerMain 27 110
POP3: server sending <example after he/she chooses some courses in Comp Science
=>
POP3: server sending <dep<BR>haresh_srm:=20>
POP3: server sending <ok<BR>jwalamalid: he can also choose some other courses
in other =>
POP3: server sending <departments=20>
POP3: server sending <too?<BR>jwalamalid: is it so?<BR>haresh_srm: yes that is
=20>
POP3: server sending <correct<BR>haresh_srm: but you should not allow<BR>haresh_
srm: more than =>
POP3: server sending <5=20>
POP3: server sending <courses<BR>haresh_srm: to be slected<BR>haresh_srm: for a=
20>
POP3: server sending <semester<BR>jwalamalid: OK<BR>saravanbbb: Wa is the min
no of=20>
POP3: server sending <courses?<BR>haresh_srm: 3<BR>jwalamalid: Actually we are
using Course =>
POP3: server sending <Catalog=20>
POP3: server sending <No. as one of the field in Course Catalog table<BR>jwalam
alid: what is =>
POP3: server sending <use of=20>
POP3: server sending <the Course Catalog No?<BR>haresh_srm: okcourse catalog num
ber is just a=20>
POP3: server sending <hbreviationg of trhe fill course name<BR>haresh_srm: Advan
ced Java =3D =>
POP3: server sending <Course=20>
POP3: server sending <Name<BR>haresh_srm: CS898 =3D Course Catalog<BR>saravanbbb
: The watz the =>
POP3: server sending <diff bet=20>
POP3: server sending <course catalog no and the course id?<BR>haresh_srm: i dont
know what's =>
POP3: server sending <course=20>
POP3: server sending <id<BR>haresh_srm: there is a section nnumber<BR>saravanbbb
: Course =>
POP3: server sending <catalog is=20>
POP3: server sending <unique to each course?<BR>haresh_srm: yes<BR>haresh_srm: s
ometimes the =>
POP3: server sending <same=20>
POP3: server sending <course can be offered at differnet timings<BR>haresh_srm:
then it=20>
POP3: server sending <means<BR>haresh_srm: there wil be differnet section =>
POP3: server sending <numbers<BR>haresh_srm: i=20>
POP3: server sending <guess i mentiodn this before during<BR>haresh_srm: the=20>
POP3: server sending <requirements<BR>jwalamalid: Yeah<BR>saravanbbb: Okay we
got=20>
POP3: server sending <that<BR>haresh_srm: cool<BR>saravanbbb: Pls send us the de
t regarding =>
POP3: server sending <CGPA=20>
POP3: server sending <ASAP<BR>jwalamalid: Thanx for ur answers<BR>haresh_srm:
=>
POP3: server sending <ok<BR>haresh_srm:=20>
POP3: server sending <sure<BR>haresh_srm: tomorrow<BR>jwalamalid: we will mail
u the DB=20>
POP3: server sending <2day<BR></FONT></DIU></BODY></HTML>>
POP3: server sending <>
POP3: server sending <-----=_NextPart_000_005D_01C52ED7.7695C3E0-->
POP3: server sending <>
POP3: server sending <.)
POP3: client sent [DELE 1]
POP3: server sending <+OK>
POP3: client sent [QUIT]
POP3: entering update state
mailbox deleted 1 messages
POP3: finished session, closing connection
```

B-2 DETAILS OF THE MESSAGE