INTRANET MAIL SERVER

PROJECT WORK DONE AT
WINSOFT TECHNOLOGIES (P) LTD

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF
BACHELOR OF ENGINEERING –
COMPUTER SCIENCE AND ENGINEERING
BHARATHIAR UNIIVERSITY, COIMBATORE.

P- 665

SUBMITTED BY

Mr. C. Deepak Chandran (9827k0170)

Mr. Parag. M. Maniar (9827k0193)

Mr. R. Balachander (9827k0717)

Mr. B. Saravanan (9827k0720)

P. 665

GUIDED BY

External Guidance
Mr. Mubarak Musthafa
Winsoft Technologies,
Coimbatore.

Internal Guidance
Mrs. S.Devaki, B.E, M.S.,
Assistant Professor,
Dept of Computer Science,
Kumaraguru College of Technology,
Coimbatore

Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore - 641 006

March 2002.

Winsoft Technologies

on time every time

Date: March 02, 2002

CERTIFICATE

This is to certify that this project work entitled **INTRANET MAIL SERVER** being submitted by

Mr. Parag.M.Maniar [Reg.no.9827K0193]
Mr. Deepak Chandran [Reg.no.9827K0170]
Mr. R.Balachander [Reg.no.9827K0717]
Mr. B.Saravanan [Reg.no.9827K0720]

In partial fulfillment of the requirements for the award of the degree **Bachelor of Engineering** in Computer Science and Engineering is a bonafied record of the work carried out at **Winsoft Technologies**, Coimbatore, under my supervision and guidance.

During their project period their behavior and attitude towards learning were excellent. They were also very sincere and hard working.

For **Winsoft Technologies**

Mubarak Musthafa
Director

CERTIFICATE

Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore - 641006.

This is to certify that the project work entitled

Intranet Mail Server

Done by

Mr. C.Deepak Chandran (9827k0170)
Mr. Parag.M.Maniar (9827k0193)
Mr. R.Balachander (9827k0717)
Mr. B.Saravanan (9827k0720)

In partial fulfillment for the award of the degree of
Bachelor of Engineering in Computer Science and Engineering
of Bharathiar University, Coimbatore
during the academic year 2001 - 2002

12/3/02

Professor and Head
Dr. S.Thangasamy, B.E, Ph.D.,

Guide
Mrs. S.Devaki, B.E, M.S.,

Certified that the candidate was examined by us in the Project Work Viva Voce Examination held on ..... and the University Register Number was .....

Internal Examiner

External Examiner

Acknowledgement

The endeavor over a long period can be successful by the advice and support of many well-wishers. We avail this opportunity to express our gratitude and appreciation of all of them.

We express our profound respect and sincere gratitude to our Principal **Dr. K.K.Padmanaban, B.Sc. (Engg), M.Tech, Ph.D.,** for having provided the necessary facilities to complete this project.

We are greatly indebted to our Head of the Department **Dr. S.Thangasamy, B.E (Hons), Ph.D.,** Computer Science and Engineering, Kumaraguru College of Technology, for our source of inspiration and encouragement rendered by him.

We are greatly privileged to express our deep sense of gratitude to our guide **Mrs. S.Devaki, B.E, M.S.,** Asst.Professor, Department of Computer Science and Engineering, who has been a motivating force behind all our deeds.

We wish to extend our gratitude to **Mr. Mubarak Musthafa**, Winsoft Technologies, Coimbatore for allowing us to carry out this project work at his concern and guiding us to complete this project successfully.

Last but not the least, we wish to thank all our friends for their continuous support and encouragement during the course of this project.

Synopsis

Synopsis

E-mail has become the primary medium of communication worldwide. **INTRANET MAIL SERVER** is a scalable, reliable and secure Email server. All the basic requirements can be met in a LAN using this powerful full-fledged Server.

INTRANET MAIL SERVER is designed to meet all the basic facilities in LAN. It has the features of sending mails, receiving mails within an intranet, and also through Internet, an event reminder (calendar) common for all the users, workflow modules for the automatic transferring of messages in a specified order of priority, and finally chatting facilities with both public and private chat within the LAN. The project is implemented using ASP and MS-ACCESS.

Goals

The **INTRANET MAIL SERVER** has envisaged in a better and broad way. It is designed and developed to have more facilities, which meet the requirements of the company, the user and the administrator, which are described below.

For the Company

- Capable of sending mails and receiving mails on **LAN.**
- Capable of sending and receiving mails through **Internet.**
- Use of **workflow** as per the company's preferences for any organization.
- Complete **chatting** with both the public and private chat.
- An **event remainder** for all the users.

For the User

- All the users are provided with unique id and password.
- Users can send and receive messages anywhere within the LAN.
- No limit on size of mails.
- Users can send their letters through the Workflow, which directs it automatically in a pre-defined order.

For the Administrator

- This provides administrator with easy to use GUI interface.
- Administrator can add new users, delete or modify users.
- It is not possible for more than one user to login with same id, simultaneously.

International standards

As the application is coded in **ASP** it can work on any system that supports windows and the Internet explorer.

Contents

CONTENTS

Chapter 1

Introduction

Introduction

1.1 Company Profile

Winsoft Technologies is a 5-year old business solutions provider with extensive experience in building web-based applications, Shopping carts, Sales Automation systems and Client-Server Applications.

Strong technology skills are the defining USP of **Winsoft**. The company has a strong team of engineers and architects working on the Microsoft and java platforms. Typical projects involve advanced n-tier architecture, which integrates with backend legacy systems and databases. Winsoft has a dedicated team working on .net compliant high-end application servers like Microsoft small business servers and site servers. A team of designers and graphic visualizes excel in creating light-weight home pages, creative flash demos, intuitive navigation and user friendly forms and screen layouts. The **Winsoft QC team** ensures that all development is bug-free and adhering to the specifications.

The team is committed to excellence and earning profits for its stakeholders without compromising on corporate ethics. **Winsoft Technologies** has executed over 10 large-scale projects for clients across India including Multi-nationals and technology companies.

The current **domain expertise** of Winsoft extends to several verticals including finance, hospitality, printing, sales force automation and market places. Legacy system integration and EAI is the current focus and the engine for future growth.

The development methodology is custom designed to handle offshore development. The methodology defines streamlined processes for application design, technology development, workflow and communication with our customers.

1.2 Existing System and its limitation

Electronic communication, because of its speed, turnaround time and broadcasting ability, is exploding the world much faster than expected by industry experts. Sending mails has become primary concern not only in the Internet but also within intranet. Many corporations use Intranet, instead of Internet, to offer their employees easy access to corporate information, while preventing outside access to that data.

Let us examine how an organization can provide with the facilities of serving messages and various other intranet facilities to its employees.

The use of chatting has become very common and feasible in our day-to-day life. But in an organization were people are recommended to be in touch with others, chatting place a vital role within the organization. Chatting doesn't only play role in the providing facility of public chat but also must enhance the use of private chat.

The workflow flow plays vital roles in any organization were there is need to show any legal documents to higher authorities. The Workflow management will take care of forwarding all the messages to the higher authorities in a streamline path, by having set priority levels to various persons in an organization.

Limitations

The limitations of the existing system are...

- No privacy of individual mails. No user, within an organization can feel that his privacy is maintained.
- Requires a machine that is dedicated for this purpose. Not all users can be facilitating with the dedicated system.
- No two users can log in with same and access the same work area.
- Acknowledgement of further send/received mails cannot be received.
- The organization may not have any control over the system.
- Restrictions or constrains of the system cannot change with the change in policy of an organization.

These are the series limitations that are faced when an organization is not able to provide its facilities to all its users. Both the organization and the employees have their own pros and corns on the limitations.

1.3 Proposed System and its advantages

The primary objective of INTRANET MAIL SERVER is to provide mail facilities within the LAN including the added facilities of chatting, and event remainder tailored with workflow management. The modules of the proposed system are...

The authentication of the users with a password for using the various facilities helps to incorporate strong security within the LAN. This authentication also takes care that only a single user is logged in with his/her particular id, at any specific time.

(I) Sending and Receiving mails within Intranet

When the users are granted privileges to use the facilities user can send mail to particular user if that recipient is present in the organization with that user id. The message is instantly transferred to the users mailbox if the recipient's user id exists otherwise the sender is reported by appropriate errors. The recipient can also delete the unwanted mails/viewed mails.

(II) Mailing through Internet

When the users are granted privileges, it is possible to send mails with attachments, via Internet to the users present outside the organization. Mails can also be received in the same users Inbox.

(III) Intranet Chatting

The chatting facility helps the user to send instant messages online over the LAN. Here again the user is authenticated by the same password. The screen displays the list of online users by which the user can send instant messages. The users can also view the commands using the help file, and also prevent a particular user from sending messages, change the displaying speed (refresh rates), also see the credits, information of other user. The private chat is also possible between two users.

(IV) Event Remainder

The event remainder helps the users to add any events that are scheduled to takes place in near future. This event remainder also helps to recap the events that had taken place in past. The user can view all the events present in the event remainder and also add, delete or modify the texts.

(V) Work Flow Management

The workflow management is built appropriately to suite any organization. Any person in the organization can send their legal forms for verification to their higher authorities and can be replied with their status. The authorities can be ranked in the precise manner.

Advantages

1. NO dedicated system is required since the project aims on the user authentication.
2. No user can use a single work area simultaneously.
3. System is provided with auto response.
4. Review of already held events could be done using event remainder.
5. The flow and order of work can be changed dynamically.

Chapter 2

System Requirements

System Requirements

2.1 Product Definition

2.1.1 Problem Statement

The product is concerned with the sending and receiving of mails within an intranet with added facilities of chatting and event remainder, workflow management, all with respect to posting of messages in one form or other, thus named **INTRANET MAIL SERVER.**

2.1.2 Functions to be provided

The various functions to be provided to the users are the sending and receiving mails, chatting within the organization with various facilities, workflow management used for automatically sending mails to other designated authorities, the event remainder which helps the user to keep track of daily events.

2.1.3 Processing Environment

Hardware Specification

The proposed system is being implemented on the following hardware configuration.

Minimum configuration

Processor	- Pentium I
Speed	- 600 MHz
Main Memory	-333 MB
Monitor	-800x600 at 16-bit true color

Recommended Configuration

Processor	-Pentium III
Speed	- 750 MHz
Main Memory	- 512 MB
Monitor	-800x600 at 32-bit true color

Software Specifications

Scripting Language	- ASP
Tools Used	- Visual Interdev
Browser	- IE 5.0

Operating System

Windows NT/ 2K/ XP running Internet Information Service 5.0

Database

Microsoft Access

2.1.4 Solution Strategy

The problem was approached in a step-by-step fashion. First and foremost the functions and their purposes are learnt. Then analysis is done on problem definition. Then the rough draft is made on the analysis and finally ends up in a solution by breaking down the defined problem into four modules.

The solution is to have a Link option as the Interface for this product. The user can choose the desired link at any time of instance and the module is automatically called.

2.1.5 Product Features

The main feature of this package is User-friendly and authentication packed package. Verifying the name and password of any user who accesses the company checks it. Another important feature of our project is the possibility of private chat within public chat. The workflow management order and access can be formed and changed dynamically with the requirements. The requirements of the customer are studied and are met which ultimately leads to customer satisfaction. The response time is less and it is user friendly too.

2.1.6 Acceptance Criteria

The input given by the user who accesses the company is his name and password, which is verified with the database to check for authentication. Then as per the requirements the user is facilitated as per the order and priority of the user. Each user has their own level of access to the system, depending on the acceptance criteria are fixed by the administrator.

2.2 Project *plan*

Life Cycle Model

The Spiral Model is proposed to be life cycle mode followed while developing the software. It provides the potential for the rapid development of incremental version of the software. The software is developed in the series of incremental releases. The spiral model has six tasks region.

Task Region 1:

- Terminology : Customer Communication
- Work Product :

The software programmer defines about their existing intranet facility and the facilities that are to be incorporated in their Intranet server. Reading some books related to the process of communication between two systems and its use. The software must be able to support various facilities within the Intranet.

Task Region 2:

- Terminology : Planning
- Work Product : Analysis of the product definition.

What function the product has to Perform, Processing environment, Product features, Programming Language and development tools to be implemented are all decided in this stage.

Task Region 3:

- Terminology : Risk Analysis
- Work Product :

Technical Risk:

Since Design Document plays a very vital role in coding, Preparation of Design Documentation consumes more time. Therefore coding is based on how we design the screen pages and the various links, what are all the logics, concepts that can be used. Once the design is made the coding can be started from scratch without any confusion.

Managerial Risk:

For every module time limits are set to be fixed for its completion. The project duration is between Nov – March. With in the given time slot for each module it has to be completed. The modules cannot be postponed because of the construction of various individual modules. If there is a delay in any one of the modules there is a delay in completion of the whole project. Therefore the modules have to be completed in the appropriate time as per the schedule.

Task Region 4:

- Terminology: Engineering.
- Work Product: SRS

Based on the needs, the software requirement specification is prepared. SRS includes Product Overview, Processing Environment, External interface and dataflow, Functional Specifications, Performance requirements, Exception condition & handling, Early subsets, Foreseeable modification, Acceptance criteria, Design guidelines.

- Work Product: Design Document.

Based on the needs, the Design document is prepared. Designing plays an important role during coding. Once the design is framed well, the programmer can start the coding very easily. Design document includes external design specification, Architectural design overview and Detailed design Specification.

Task Region 5:

- Terminology : Construction and Release.
- Work Product :

* Front End Design like adding buttons, text boxes with various images and gif images Various attaching handlers to link options and calling its corresponding boxes or its functions

* Coding for all the links and text boxes that accept input through edit controls.

* Coding for connectivity with the database and the forms that loads and saves in the database.

* All the transactions between the forms and the database are done through coding.

* Preparing user Manual.

* Preparing Test Plan.

* Demonstrating the whole project.

Task Region 6:

- Terminology : Customer Evaluation.
- Work Product :

Customer Evaluation is that the feedback of the customers after seeing the partial or final outcome of the product. Even then also the product may be undergone for change, if the customer is in need of.

2.3 Scripting Language and development tools

Scripting Language

Scripting languages are an intermediate stage between languages that deal with layout of a page such as HTML and VRML and programming languages such as Java, C++ and Visual Basic. The ASP technology and Microsoft IIS are arguably the hottest products ever for building Internet, Intranet and Extranet applications. ASP has quickly become the standard for RAD environments.

ASP (Active Server Pages)

ASP is a server based scripting language that is used to build database driven website were the browser may have no scripts at all. ASP scripts can be executed off line that can tested using PWS or IIS.ASP scripts are pure ASCII and can be edited with notepad and more sophisticated tools like Visual Interdev. These codes are not biased towards any browser. Scripting can be written in Jscript, Java Script and Pearl Script, it also supports server components build with other languages. ASP provides

- A way to save individualized data for each user.
- Access to the file system.
- Access to databases.
- A means to launch and control any Component Object Model component.

P. 665

VBScript

By default ASP scripting language is set as VBScript. It can communicate with host applications using ActiveX technology. The main advantage is its simplicity.

JavaScript

Java Script being widely used has also become the standard scripting language. JavaScript and Jscript are interpreted object oriented languages. JavaScript is a loosely typed language therefore it not necessary to declare the variables explicitly.

Database

Being user-friendly to use, MS-Access acts a primary data store to ASP. Access can be accessed from ASP using common SQL-Queries. Being Microsoft product doesn't need any external connectivity drivers to connect.

Documents to be Prepared

It is suggested that the following documents can be prepared during the time of the project.

- A System Definition consisting of a Product Definition and a Project Plan.
- A Software Requirements Specification.
- A Design document consisting of external design, architectural design, and detailed design specification.
- A test Plan.
- A User's manual.
- A properly documented, debugged, and tested program.
- A project legacy document.

Chapter 3

Software Requirement And Specifications

Software Requirements and Specifications

3.1 Introduction

3.1.1 Scope

The primary scope of **INTRANET MAIL SERVER** is to provide mail facilities within the LAN including the added facilities of chatting, and event remainder tailored with dynamic flow and access workflow management and also to provide Privacy and security features between the users.

3.1.2 Overview

INTRANET MAIL SERVER is designed to meet all the basic facilities in LAN. It has the features of sending mails, receiving mails within an intranet and also through the internet, an event reminder common for all the users $, workflow modules for the automatic transferring of messages, and finally chatting facilities with both public and private chat within the LAN. It has envisaged in a better and broad way. It is designed and developed to have more facilities, which meet the requirements of the company, the user and the administrator.

3.2 General Description

3.2.1 Product Function

The various functions to be provided to the users are the sending and receiving mails, chatting within the organization with various facilities, workflow management used for automatically sending mails to other designated authorities, the event remainder which helps the user to keep track of daily events.

3.2.2 User Characteristics

Only the employees working in a particular organization can use this product. The software is designed with user-friendly interface. The user can interact with the system by using several link options for input and also for selection of desired facility with appropriate validations.

The system requires an administrator to configure the system who is the sole authority. Administrator is responsible for adding, modifying and deleting users and flow. The end user is requested to know only to send and receive in the system.

3.2.3 User Problem Statement

The product is concerned with the sending and receiving of mails within an intranet with added facilities of chatting and event remainder, workflow management, all with respect to posting of messages in one form or other, thus named **INTRANET MAIL SERVER.**

3.2.4 User Constraints

User constraints are that passed to the design phase from requirements phase. Things that should be accomplished like

- No two users with same id can login.
- Dynamic change in order and flow of work.
- Multi-utility chatting system for intranet.

3.3 Functional Requirements

The software needs to be working all through the day. Multi-users access the database simultaneously. The same record may be

access by many users. The design of the database should support these criteria. No special requirements for performance of the software are required. The administrator should govern proper login and logout of the users. Exception handlers should be properly initialized.

3.4 Exception Conditions/ Exception Handling

While programming there is a chance to come across some exceptional condition. These exceptional conditions should be handled with care; otherwise the whole software will crash. Exception conditions can occur anywhere, but most frequently occurring exception condition is discussed.

Exception Condition

If the user unknowingly leaves some text boxes blank, there is a chance for the occurrence of errors, while saving it in the database. User may not properly logout the system, multiple entries of same data.

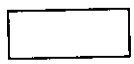
Exception Handling

These exceptions can be handled by setting the value of these fields as not null in the database or by proper notification to the user to enter properly, until a proper entry is made. A new form will be displayed if any text box is left blank, which will tell the user to retry by going back to the same form. User can redirect to same page, if duplicate entry is made. Users should be made to logout properly or the systems can automatically logout, if it senses any improper logout by the user.

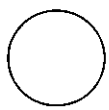
3.5 Data Flow Diagrams:

Data Flow Diagram (DFD) is a graphical technique that depicts information flow and the transforms that are applied as data move from input to output. DFD may be used to represent a system or software at any level of abstraction. In fact DFD's may be portioned into levels that represent increasing information flow and functional details.

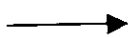
Graphical Notations used in DFD



Inputs/Outputs to the Software



Processing the inputs



Data Flows

Data Stores

Chapter 4

Design Document

Design Document

4.1 Introduction

System design is an iterative process through which requirements are translated into a "Blue Print" for constructing the software. The following are the characteristics that a design document should possess,

1. The design must implement all the explicit requirements contained in the analysis model and it must accommodate all of the implicit requirements designed by the customers.
2. The design must be readable and understandable guide for those who generate and test code and subsequently maintain the software.
3. The design should provide a complete picture of software, addressing the data, functional and behavioral domains from an implementation perspective.

4.2 Design Phase Activities

Following the guidelines the various activities that were to be carried during the design phase of the **INTRANET MAIL SERVER** are...

- INPUT DESIGN
- OUTPUT DESIGN
- DATABASE DESIGN
- PROCESS DESIGN
- SCREEN DESIGN

The whole system consists of the following modules... The description of the modules in terms of system design is...

4.2.1 Sending and Receiving mails in Intranet

The sending mail module deles with the composing and sending mail from a client to a client/many. It requires very minimal input from the administrator to begin its operation. All the inputs made to the system by the user are validated appropriately and is made in such a way that the user is optimally guided. Each user has defined privileged levels governed by the system.

Configuring the mail Server

The following configurations are absolutely necessary for mail server to begin its mail service.

1.The administrator needs to allocate a dedicated area for each user who has the rights to register and modify the users of the mail server.

2.When we start the mail server for the first time there will only be one user, namely the administrator. The administrator needs to provide a unique login id and password to the users to start the action.

Output design of S/R in intranet

There may be many users within an organization sending the mails. The mail server receives all the mails and stores in the mail Box. These stored mails are appropriately redirected to the particular Inbox of the users within the organization. The sent mails are appropriately acknowledged as soon as the receiver browses the mail. The messages that sent by the client to each other can be viewed in the Inbox by clicking the subject link. The user is redirected to view the message and if need can delete the message.

Database Design

As such the system needs considerably a good amount of storage space to store the mails collected from the users. The database design plays a vital role in the development of the system. The system creates and maintains records for each incoming and outgoing mails. The incoming mails are stored in the database **design** that acts as an inbox to the receivers. The outgoing mails are stored in the table **receive** which acts as outbox to the sender. The details of these tables are maintained and if necessary deleted by the administrator.

The table structures for each of the tables used in this module are as follows...

Inbox table

Field Name	Field Type	Description
ID	Auto	Sequence Number
From1	Text	Name of the sender
Toe	Text	Name of the Recipient
Subject	Text	Subject of the message
CC	Text	Addresses to send carbon copies
Message	Memo	The original message
Res_dat	Date/time	Date of message send
Chek_dat	Date/time	Date of message received
Status	Yes/no	1- if mail read, 0 – if mail unread

Outbox table

Field Name	Field Type	Description
ID	Auto	Sequence Number
From1	Text	Name of the sender
Toe	Text	Name of the Recipient
Subject	Text	Subject of the message
CC	Text	Addresses to send carbon copies
Message	Memo	The original message
Date1	Date/time	Date of message send

Members Table

Field Name	Field Type	Description
Username	Text	Name of the user
Password	Text	Password of the user
Email	Text	E-mail id of the user

Process Design

Software engineering process is the glue that holds the technology layers together and enables rationale and timely development of the computer software. Process defines framework for a set of key process area that must established for effective delivery of the software. The key process areas form the basis management control of software projects. The process involved in sending and receiving modules is,

The mails that is composed and send by the sender is first taken to outbox of the sender. The server makes a copy of the mail in the corresponding users inbox. When the recipient requests for the inbox,

the inbox will be displayed in the screen. By clicking the subject link, the message will be displayed. The user is set free to keep/delete the message from his inbox. As soon as the user view his message the status field will be set. The acknowledgement message composing of current date and subject will be send by the server to the sender. When the user types in the name, which is not a valid id, corresponding error is displayed.

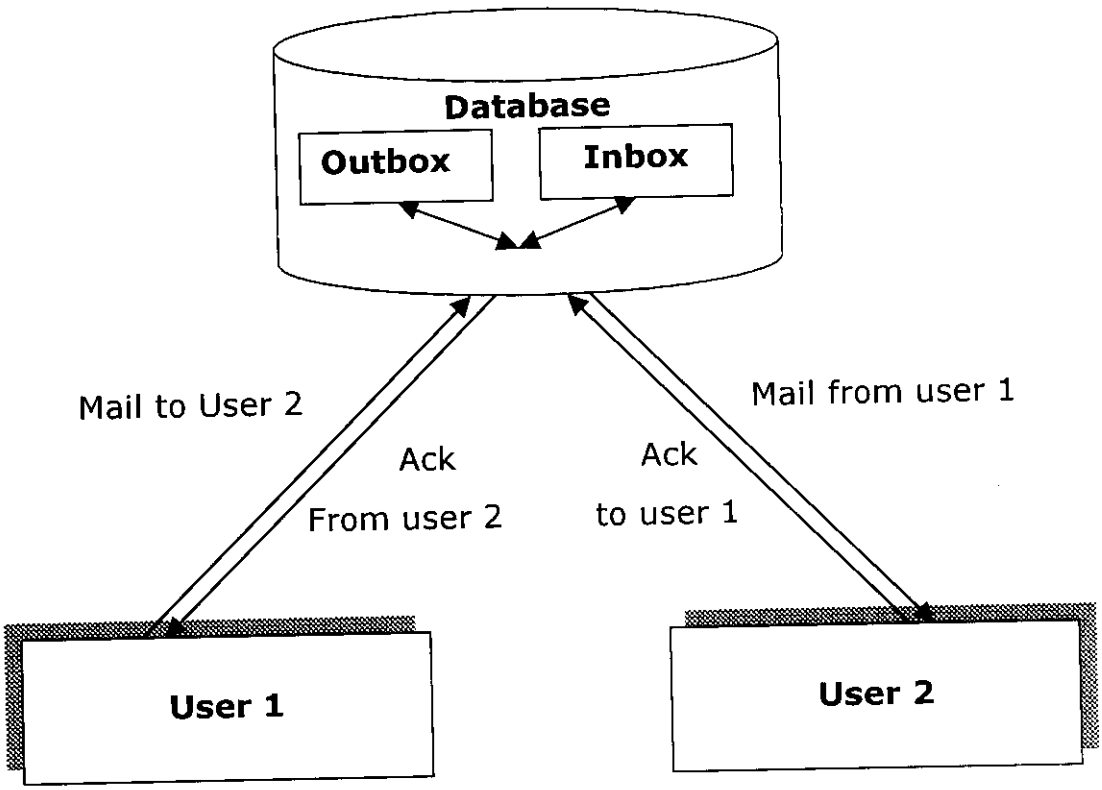


Figure 4.1

S/R within Intranet

Screen Design

The modules consist of 3 screens name compose page, inbox and message page. The compose page consist of fields like to, subject,

cc, etc. It is mandatory to specify the destination and subject fields. The inbox screen displays the inbox of the client with status. The message screen is redirected screen from the inbox, which holds the message with address.

4.2.2 Mailing through Internet

The sending and receiving mails through Internet is the module used to send/receive mails from users outside the organization. This module is same like the previous model. No separate inbox and outbox is required to store the data. The same **outbox** table is used. No **members** table is required.

Using SMTPMail itself to attach files is very easy, as all you need to do is specify the path, on the server, to the file you want to attach with the appropriate method. In many cases, the file originally resides on the workstation where you'll be sitting while you're composing your e-mail. Or, perhaps you want to create a web application that allows any user to send e-mail with an attachment. In this case, you'll need to get the file from the client onto the server before it can be attached to an outgoing e-mail. This is where Mail Server steps in to the equation.

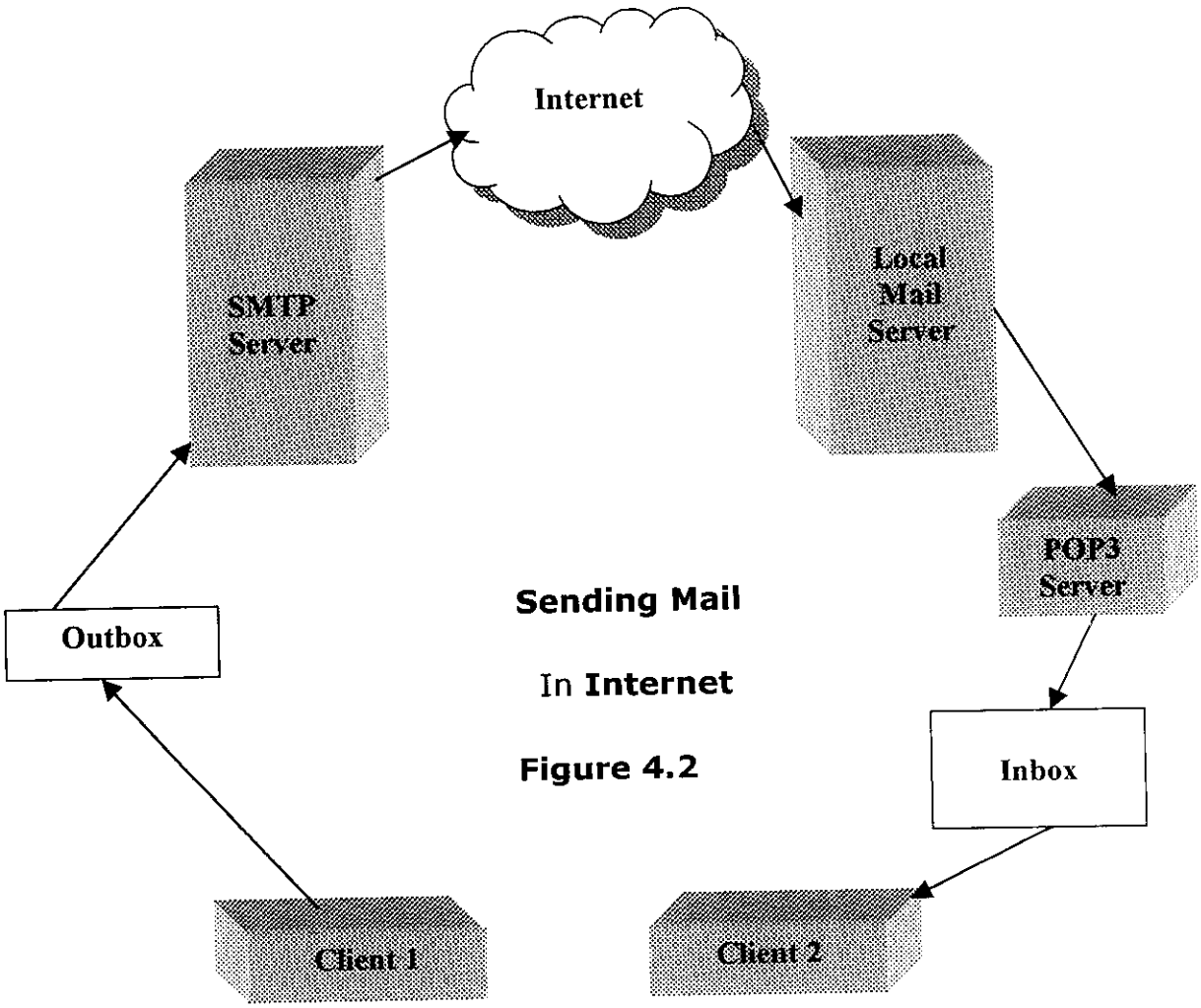
First, mail server handles receiving the file you want to attach to your outgoing e-mail. Then, it saves the file temporarily to the server's disk so that SMTPMail can attach it to an outgoing message. Finally, mail server removes the file from the server after the message has been sent, and you're done.

Configuration

The mail server needed to be configured for Internet access. It requires the details of the connection with the Internet service provider like

- SMTP Host name
- SMTP Port (default 25)

The input design is like same as the previous model. But the process of the ASP page differs. No inbox is required in this phase.



Sending Mail

In **Internet**

Figure 4.2

Sending Mails

There may be many users within organization sending the mails. The Mail Server receives the mails and stores it in the outbox of the individual users. If system is connected online to the Internet, then it will immediately send all the mails to the ISP server. If the system is not connected, then it will be stored in the outbox till it is connected. The very same way an attachment can also be send.

Process Activity

The process activity of this module doesn't require any authentication check, while your enter the TO, CC address in the compose page. When a message is composed and sent first it will be send to the pickup folder in the mailroot folder of the IIS service. The mails from here will be picked up and send to the queue folder where the mails are delivered in the queue. If the system is not connected to the Internet then the mail will not be delivered. It will be send to the badmail folder specifying failure message.

4.2.3 Chatting

The chatting module is designed for on-line communication within intranet. Generally, when we open a chatting session, many will be chatting already. We have to join the session and continue chatting. Here the chat module can be split into two things like

- Public chat: All the users who have currently logged on in this chat can view the message sent by one user.
- Private chat: message will be displayed only in the specified user screen.

Input Design

The input design of the chatting starts with the login id and password entry to the system by the user. When the verification is complete, the gets enter into the chat main page. The chat page will be split into 3, whose details can be see in the corresponding topics. In this chat page, the input is the message that we type to send it to the other users. Several options are available to enhance the features of the chat. Some of the features that are added in this module are

- Private message to a single user.
- Change the color of the user name that appears in the screen.
- Change the refresh rate.
- View the help page.
- View the details of the other users.
- Delete the message cache.

It is enough to get the chat usage rights from the administrator to access the above-specified features.

Output Design

The output design of this module is framed solely on the chat page itself. When we type-in any message at the text box available at the bottom of the chat page and submit the page immediately the message will be flashed at the mid of the page in all the user screen who have currently logged on in chat. The features that are specified in the input design can be accessed by typing the slash (/) and followed by the keyword meant for the specific feature. The keywords that are used are

/ [User name] [Message] – Used to send a message to the Specified user name alone.

/ Color [#HTML Color Code] – used to change the color of

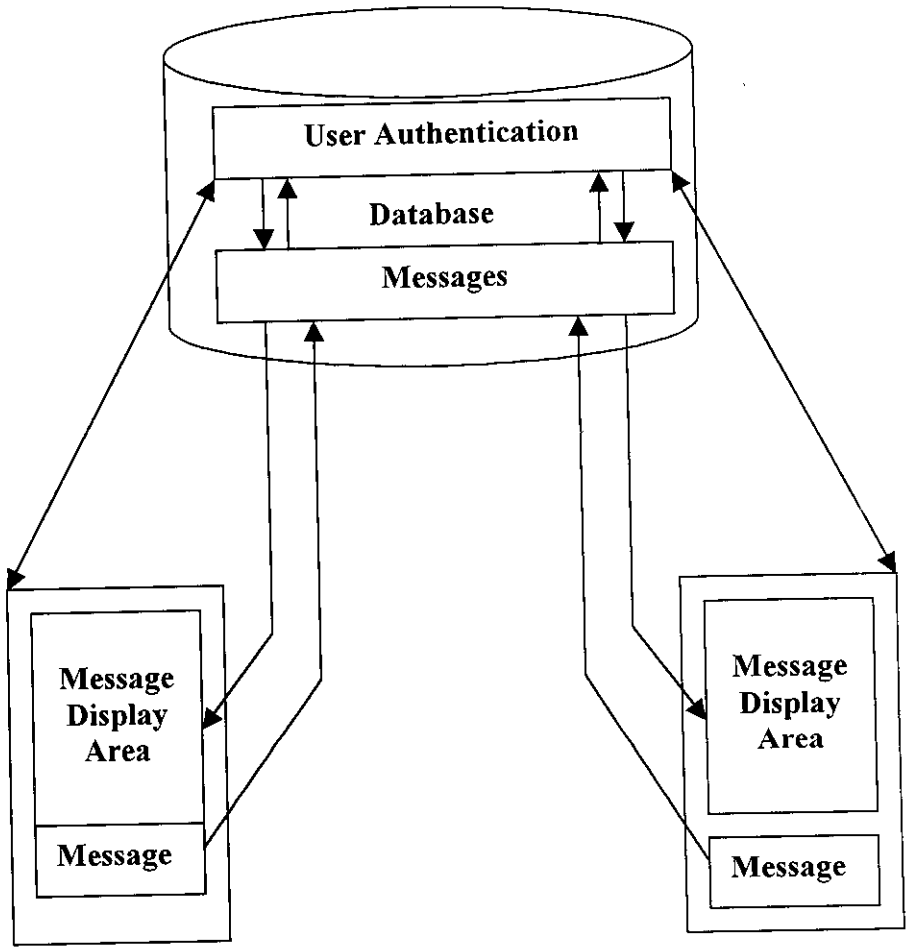
the username.

/ **Info [Name]** – Gets the information about the user name. Specified. User has to be registered to get the Information

/ **Refresh [Rate]** – Sets the refresh rate of the screen. Default is 2 seconds.

/ **Clear** – Deletes the history, message cache will be cleared.

/ **Help** – Gets the console reference of all these features.



Intranet Chatting

Figure 4.3

Process Design

The process activities involved in this module starts with the verification of user id and password. To be typed by the users at the login page, the user details will be stored already in the database. When the message is typed in the text box first the message is stored in the database and then retrieved in the message area for every n seconds specified as refresh rate. Default refresh rate will be 2 seconds, in sense that the data from the database will be retrieved automatically for every 2 seconds.

A separate table keeping track of all the logged users is maintained. The message from the database will be sent to all the users in this table. The special feature doesn't have any saved impact on the system. The features will be in effect until the user logout the session. No changes are recorded in the database.

The private chatting can be done only if the user is already logged on in the session. In the chat module also no 2 users with same ID can be login to the chat session. The administrator will take care of this privacy checks. The user can logout from the chat session by either typing the keyword /logout or by closing the browser. Both the ways will call for a logout page to inform the other users in the chat session that you have logged out from the chat session.

Database Design

The database of the Chatting module consists of four tables.

Members table

This table consists of the details about the user to login into the chat session.

Field Name	Field Type	Description
Username	Text	Name of the user
Password	Text	Password of the user
Status	Yes/no	Yes-logged on, NO-not in the chat zone

Userdb Table

This table is used to store the details of the users, called when the users uses the /info [name] command in the chat.

Field Name	Field Type	Description
Id	Auto	Sequence number
Name	Text	Name of the user
Email	Text	Email id of the user
Age	Text	Age of the user
Hobby	Text	Hobbies of the user

Logdb table

This table is used to store the name of the users who were currently chatting.

Field Name	Field Type	Description
Id	Auto	Sequence number
Name	Text	Name of the user currently logged on

Msgdb Table

This table is used to store the message that is transacted between the users.

Field Name	Field Type	Description
Id	Auto	Sequence number
Von	Text	From whom the message is from
An	Text	To whom it may concern **
Date	Date/time	Date and time of message
Message	Text	Original message transacted

** - **@All** if send to all else the name of the specific user.

4.2.4 Event Remainder

The event remainder is just like a simple calendar with some extra-added feature. Itenable the users to store any important events to be remembered. This feature is currently common to all the users. Any user can add or delete the event from the event remainder.

The input that can be providing to the event remainder is the event alone. These events can be viewed in the calendar format. The subject or the keyword will appear in the corresponding date of the calendar. By clicking the subject found in the calendar the details about the particular event will be displayed along with date and time also the name of the user who added this event in the remainder.

These remainder, has also it's own database to store the events that are stored by the users. The processing or transaction between the web page and the data stores are very direct. On click of the event the details about the event will be displayed.

Calendar Table

The table consists of the following fields

Field name	Field type	Description
Id	Auto	Sequence number
Subject	Text	Subject/keyword of the event
Message	Memo	Original message /event details
Day	Number	Day of the occurrence of event
Month	Number	Month of the occurrence of event
Year	Number	Year of the occurrence of the event
Added by	Text	Name of the user who add the event
Dateadded	Date/time	Date/time of event added

4.2.5 Workflow

The workflow is the next module in the intranet mail server. The Administrator plays the important role in this module. In the other modules the function of the administrator is to add/delete user details in the database. The second function is to provide user authentication when the user log's in.

Administrator

Three fourth of the workflow functions are done by administrator who alone has the sole rights. The user has only the one-third part of access to the workflow. Some of the functions of the administrator are as follows,

- Add/delete a new user.
- Add/delete a user group.
- Add/delete user group mapping.
- Design a new form.
- Modify/delete the existing form.
- Add/delete workflow access to the user.
- Add/delete workflow order to the form.
- View user group mapping.
- View form.
- View workflow order.

Apart from these things the administrator will take care maintaining the database interactions with the front forms.

Add/deleting a new user is done in the same way as in the other modules. Storing all the user details in the table and at the time of login, these values are retrieved to verify the login details.

Add/delete a user group is that a new category will be created and also the priority level of the newly added category will also be re ordered according to the currently available ones. For example Lecture is a designation in which many persons will be available. When a new user is to be registered the registration will also involve the category. Deletion of one category will lead to rearrangement of whole order, which will be done automatically.

Add/delete user group mapping involves assigning one user to many groups. Any user in the organization should be at least in one group. For multiple group assignment these group mapping is used.

Forms

Form is a usual page created by the administrator with some text boxes and labels. These are the forms to be filled by the users and went to the higher officials for approval. It is enough if the users just fill in and send. All the mappings will be done in the workflow order. These forms can be created, modified and deleted by the administrator. The administrator should be a liable and skilled person to operate all the functionalities.

Workflow

The workflow starts with the workflow access permissions and the workflow order. The workflow access is that the rights given to the users to access the specified forms. The workflow order is the order in which the form is to travel in the system. For example, when a student sends a leave letter it will be forwarded to the HOD through the class advisor for approval. So

Forwarding: Student – Class Advisor – HOD

Reply: HOD - Class Advisor - Student

All these setting can be view by the administrator to crosscheck the setting.

Users

The users part of this workflow is to access the forms created by the user to transfer the message in the per-defined priority order specified by the administrator. The user can view the status of the mail sent at every point of time.

Process Design

The process design involves the flow of work. Whenever the form is posted it should be properly directed with appropriate status feedback. Like in the above example when the leave letter is posted by a student the status of the leave letter should be sent to the student and also to all the persons who were involved in the flow.

Some of the status involved in the leave form submission is

- Whether Class Advisor viewed the form.
- Whether Class Advisor forwarded with/without verification or rejected
- Whether the HOD viewed the form.
- Whether the HOD has approved/rejected and replied.

The reply from the HOD will be found in the inbox of both the Class Advisor and the student. The message can be viewed by clicking the subject field of the inbox table. The inbox is the same that is available for the reception of the emails in the intranet.

Database Design

The database design consists of 13 tables in which two are the backup tables to store the transactions and the history of the operations.

Login Table

Used to store the login details of the users.

Field name	Field type	Description
Userid	Number	Id of the user
Password	Text	Password given to the user
Groupid	Number	Groupid to which user belongs primarily
Username	Text	Name of the user
verify	Yes/no	Yes-logged on; No-logged off

User Group Table

Used to specify the name of the group and its corresponding id.

Field name	Field type	Description
Id	Auto	Sequence number
Groupname	Text	Name of the group
Groupid	Number	Id number of the group

Workflow Table

Used to specify an id to each workflow.

Field name	Field type	Description
Id	Auto	Sequence number
Wid	Number	Id of the workflow
Wfname	Text	Name of the workflow

Workflow Order Table

Used to store the order of flow of work.

Field name	Field type	Description
Id	Auto	Sequence number
Dgid	Number	Destination group id
Wid	Number	Workflow id
Ogid	Number	Origin group id, where to start
Ugid	Number	User group id

History Table

Used to store the history of the transactions. A backup for this table is also created for future use if data in this table is also deleted.

Field name	Field type	Description
Id	Auto	Sequence number
Status	Number	Status of the workflow
Processid	Number	Id given to this particular process
Userid	Number	Sender id
Returnstatus	Number	Whether returned or not
Viewstatus	Number	Whether viewed or not
Duserid	Number	Destination user id
Forwarded	Number	Whether forwarded or not
Ouserid	Date/time	Starting user id
Wid	Number	Id of this Workflow
Odate_time	Date/time	Form submitted date and time
Vdate_time	Date/time	Work Viewed date and time

These are the different forms that are used in the workflow module.

Chapter 5

Product testing

Product testing

In the development of the software involves series production activities were opportunities of injection of Human fallibilities are enormous. Errors may begin to occur at the very inception of the process were the objectives may be erroneously or imperfectly specified, as well as [in] later design and development stages. Because of human inability to perform and communicate with perfection, software development is accompanied by a quality assurance activity. Software testing is very critical of the software quality assurance and represents the ultimate review of specification, design and coding.

Objectives of Testing

The objectives of testing are as follows:

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

Test Cases

The testing of software is a means of accessing or measuring the software to determine its quality. The area of testing is one of the key process areas in ensuring the quality the software. A detailed test case was designed after requirement stage with 100% condition coverage, boundary conditions (below, at, above), stress conditions (huge data, abnormal condition).

Testing is done to ensure that each instruction is executed at least once. All parts and branches are executed at least once. Identify critical parts, proper updation of intermediate and temporary files and finally check for tested output. Test results of invalid conditions and errors were recorded for corrections. After completion of individual module testing, integration testing is done with the whole integrated system.

Testability

Software testability determines how easily a program can be tested.

- Operability – The better it works the more efficiently it can be tested.
- Observability – What you see is what you Test. Incorrect output is easily identified.
- Controllability – The better the control over the software the greater is the optimization of testing.
- Understandability – The more the information the smarter ids the test.

Chapter 6

Future Enhancements

Future Enhancements

The primary objective of mail server is to provide privacy and security to the users. The system met the initial primary requirements but there is ample scope for the system to be enhanced/improved. Some of the feature and facilities that can be incorporated are

- Mail sending/receiving between Intranets to Internet can be incorporated with new versions of ESMTP.
- Some data encryption techniques can be added which will help in secure data transfer.
- Chatting including graphics audio/video interaction.
- Data compression techniques can be used to save the storage space.

Chapter 7

Conclusion

Conclusion

The intranet mail server has been successfully designed and implemented and tested. The software works effectively and efficient to its requirement. The system results in providing individual mail ids to all the users eliminating the need of depending on single machine for multiple users.

The administrator has sole rights over the system. Any restriction over the system can be imposed only through the administrator. The workflow can be changed dynamically depending on the requirement policies of the company. The organization can specify the user or employees who can enjoy the facility by configuring the users in the system.

Public chatting and private chatting with basic facilities are implemented with the messages stored in the database.

The server keeps track of user activities performed by the user and also top recover in case of any malfunction. Overall the system is serving the purpose, which it is intended for.

References

References

Text References

Greg Buczek, "ASP Developer's Guide", Tata McGaw-Hill-1999

Francis et al, "ASP beginners Guide", wrox, 2000

Web References

www.microsoft.com

msdn.Microsoft.com/asp

Net References

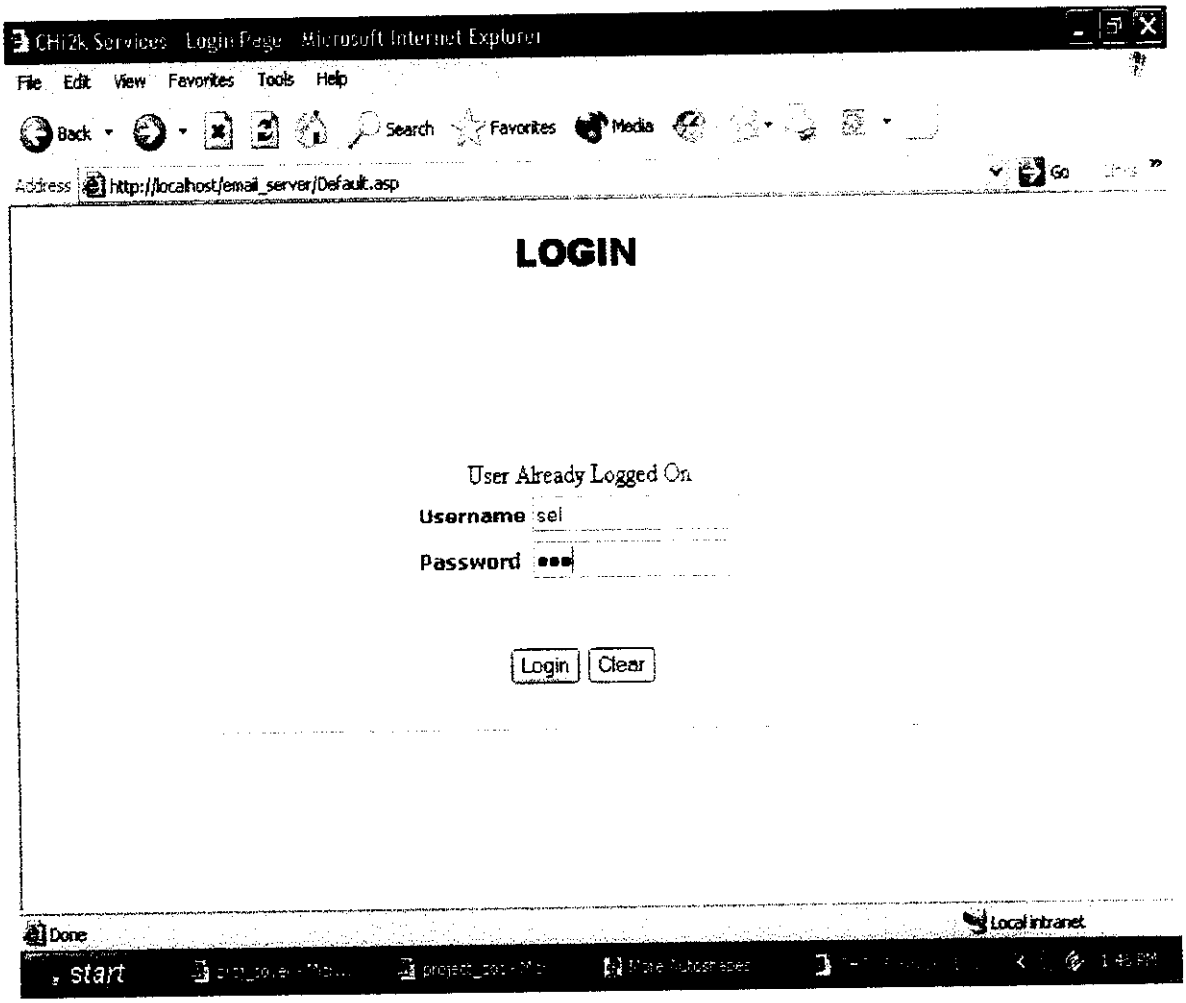
ASP Tutorials-Microsoft technical release documents

Appendix

Appendix

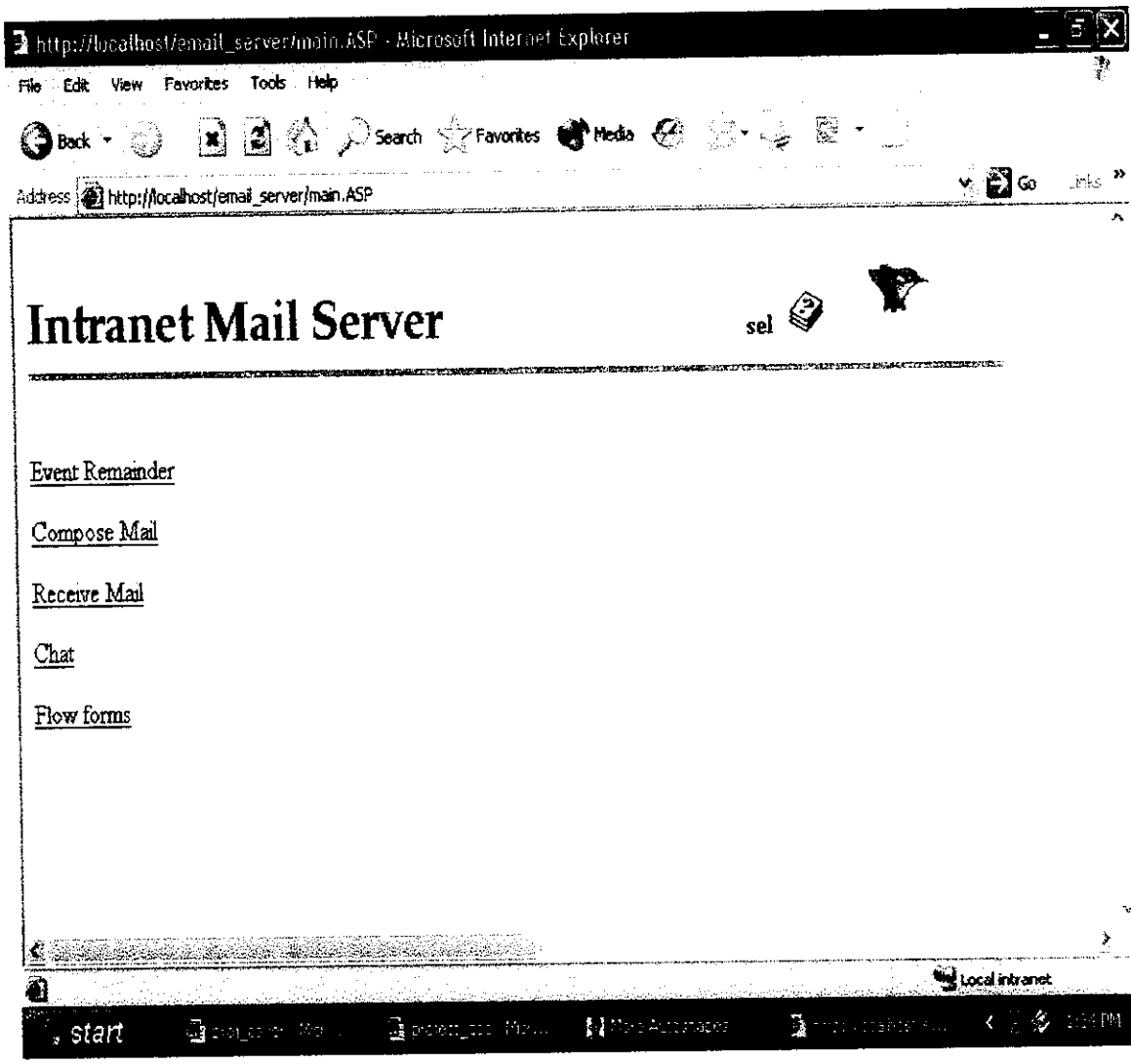
Login Page

Login page of the Intranet Mail Server, with user-built exception.



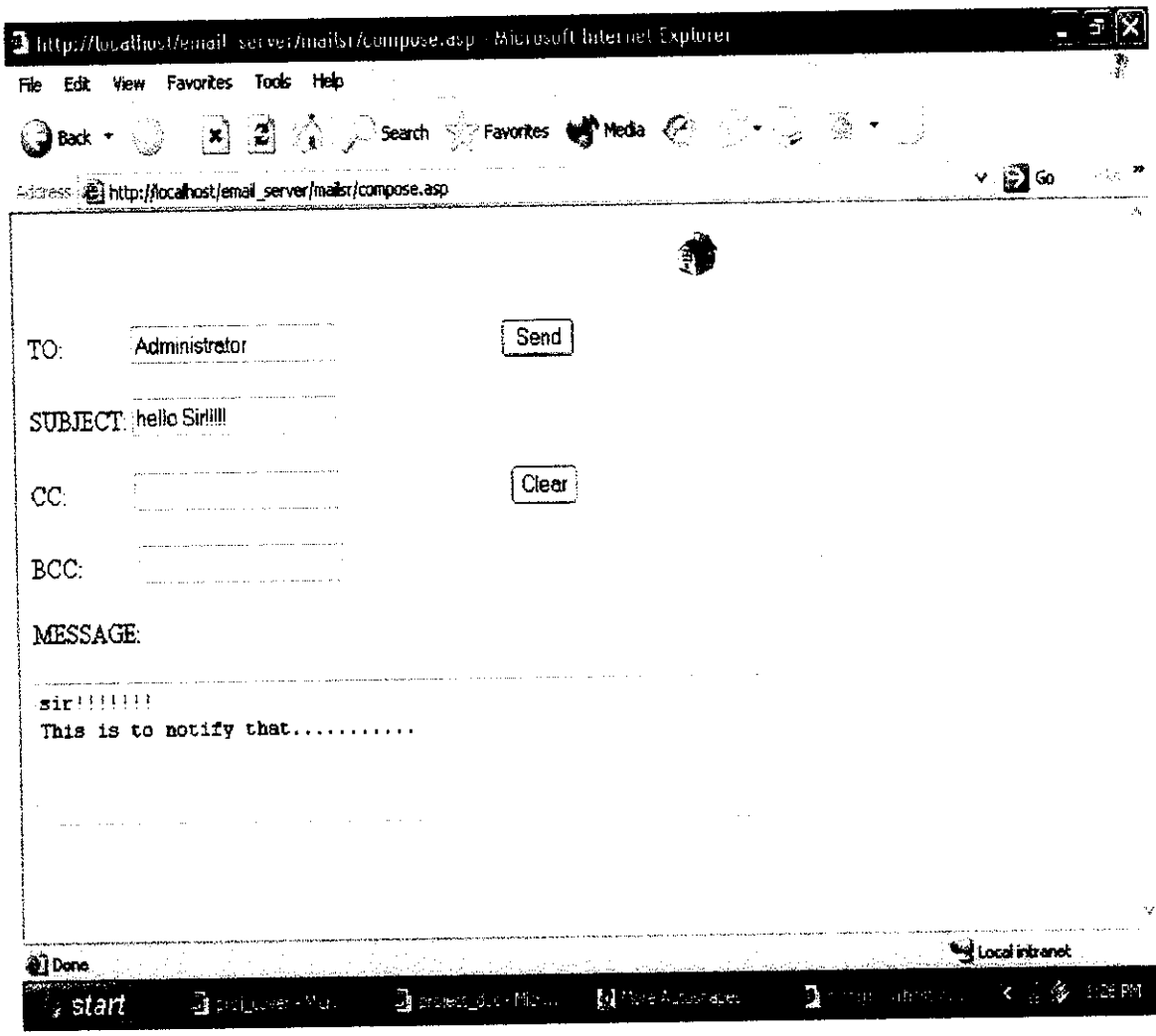
Home Page

Home page of the intranet mail server for a user with all rights except administrator.



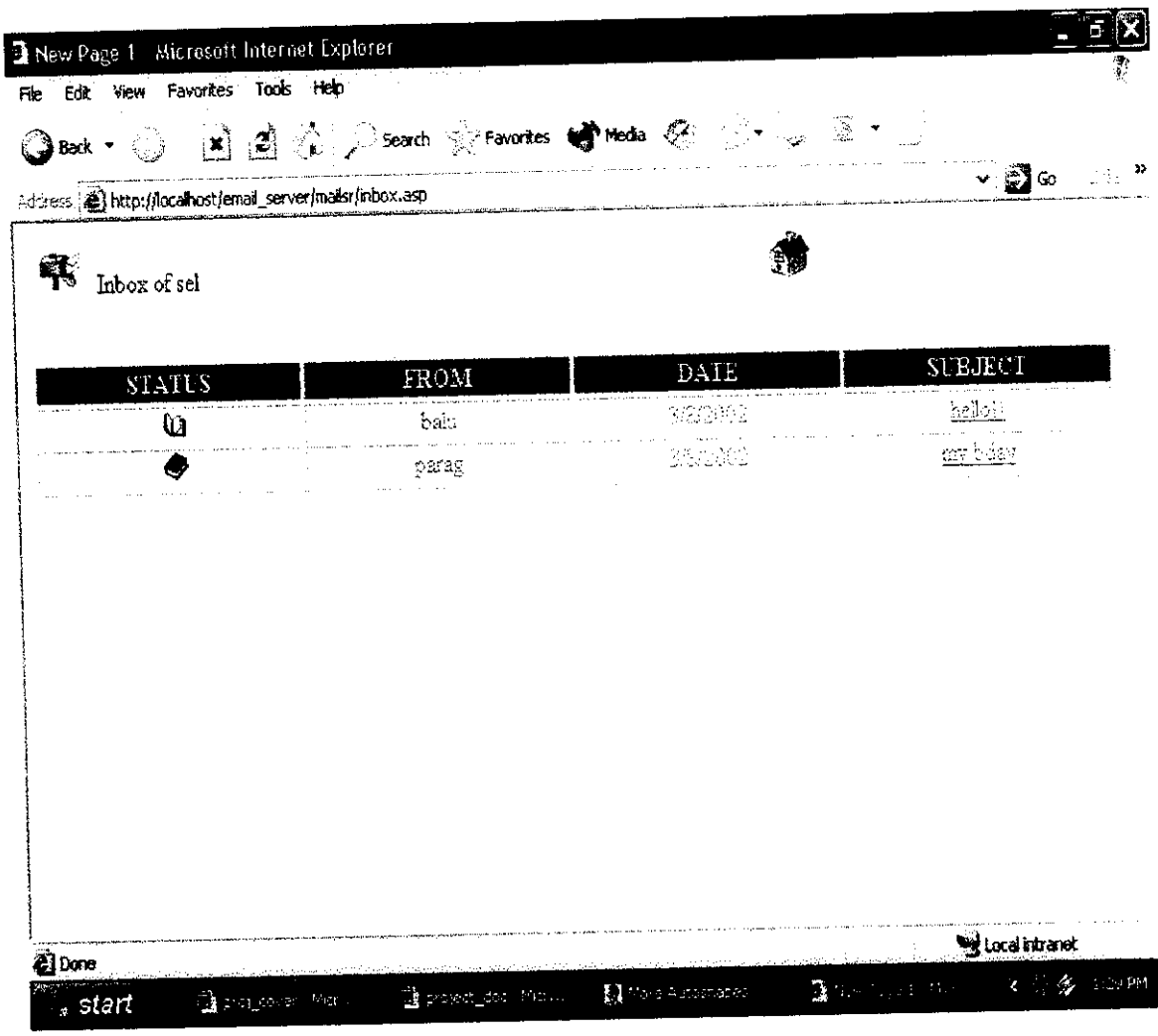
Sending Mails

Compose page of the Intranet Mail Server for sending mails within LAN



Receiving Mails

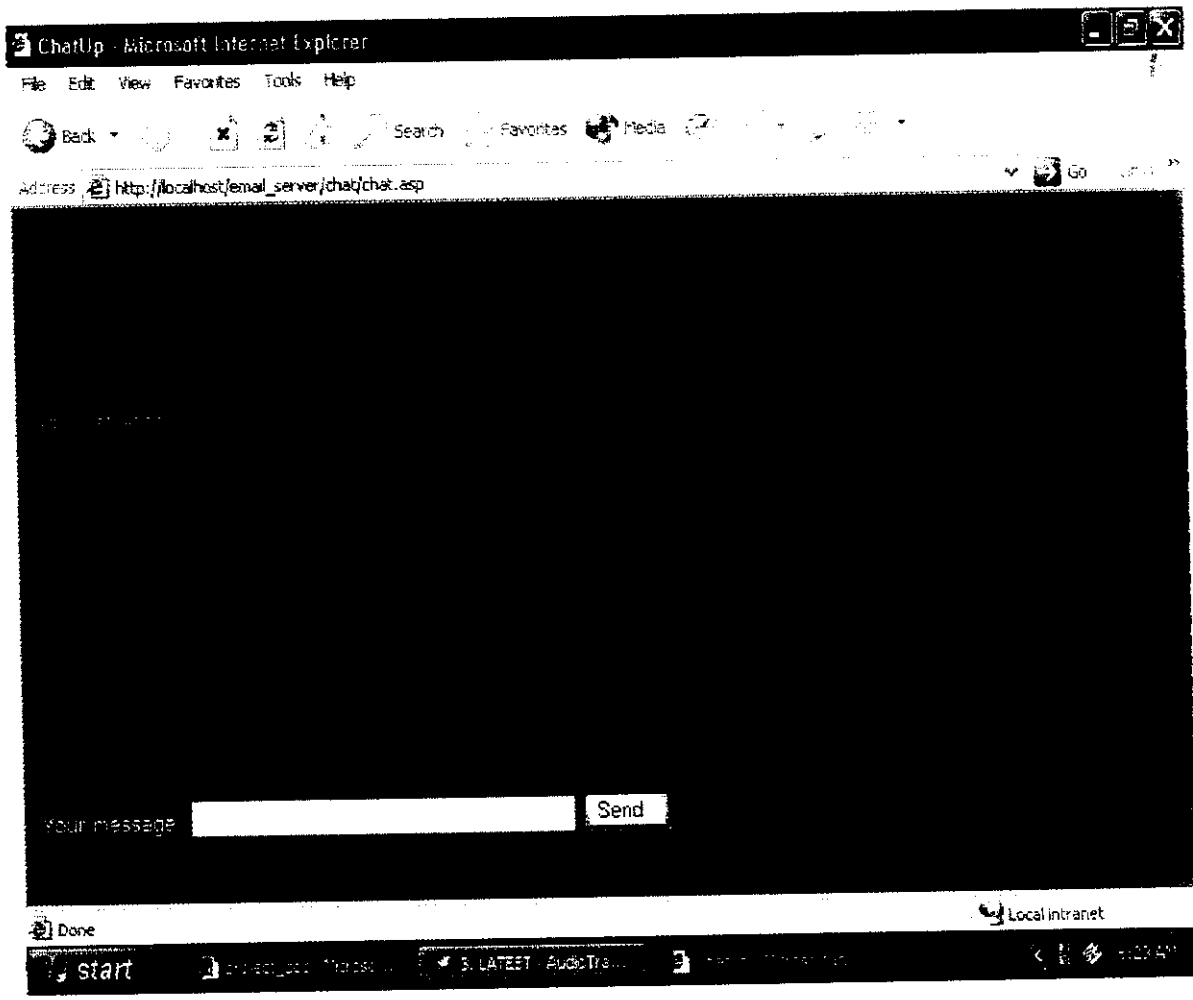
Inbox of the Intranet Mail Server, By clicking the subject hyperlink the message will be displayed.



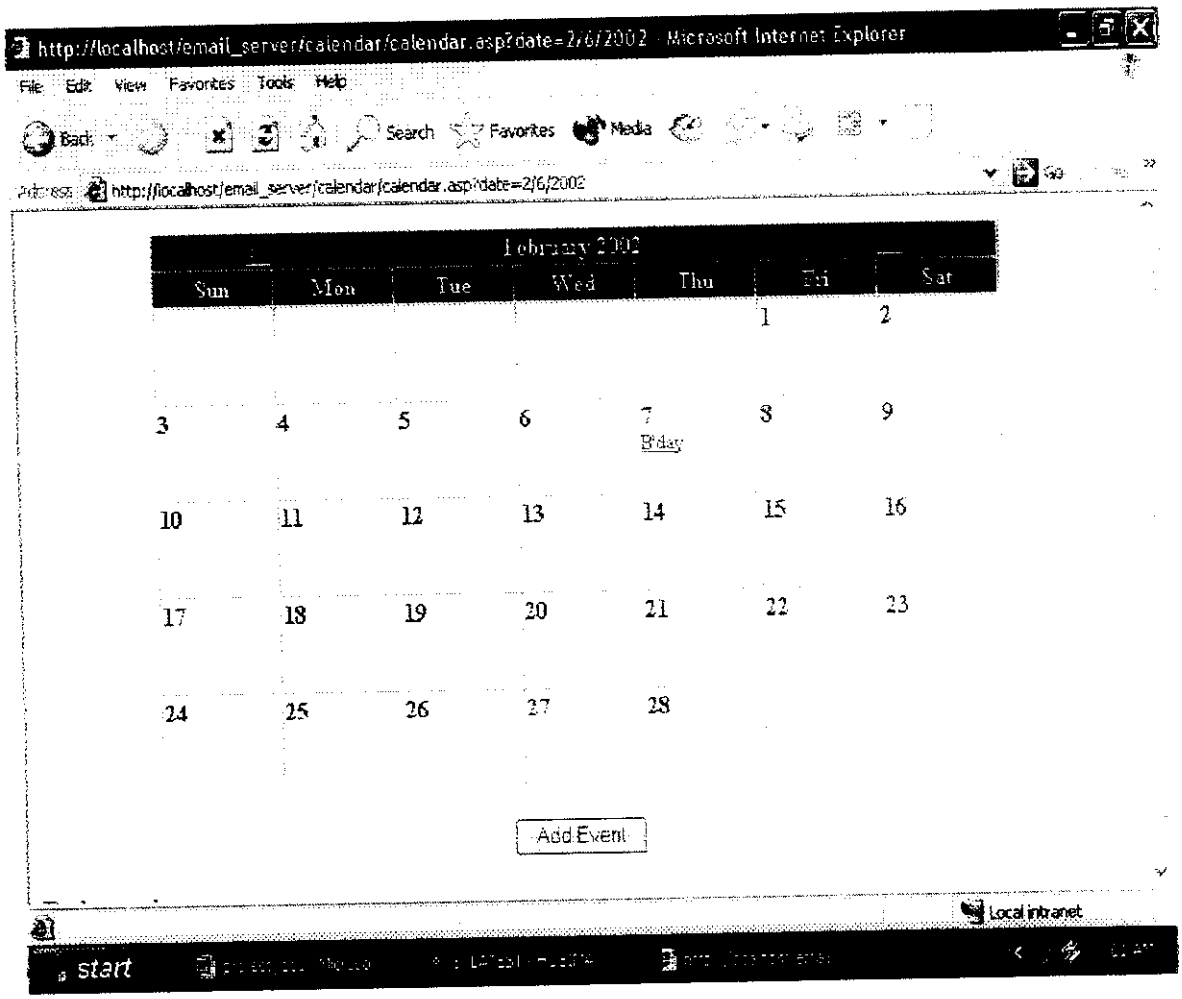
Screen of the Chat

Screen of the chat page in Intranet Mail Server.

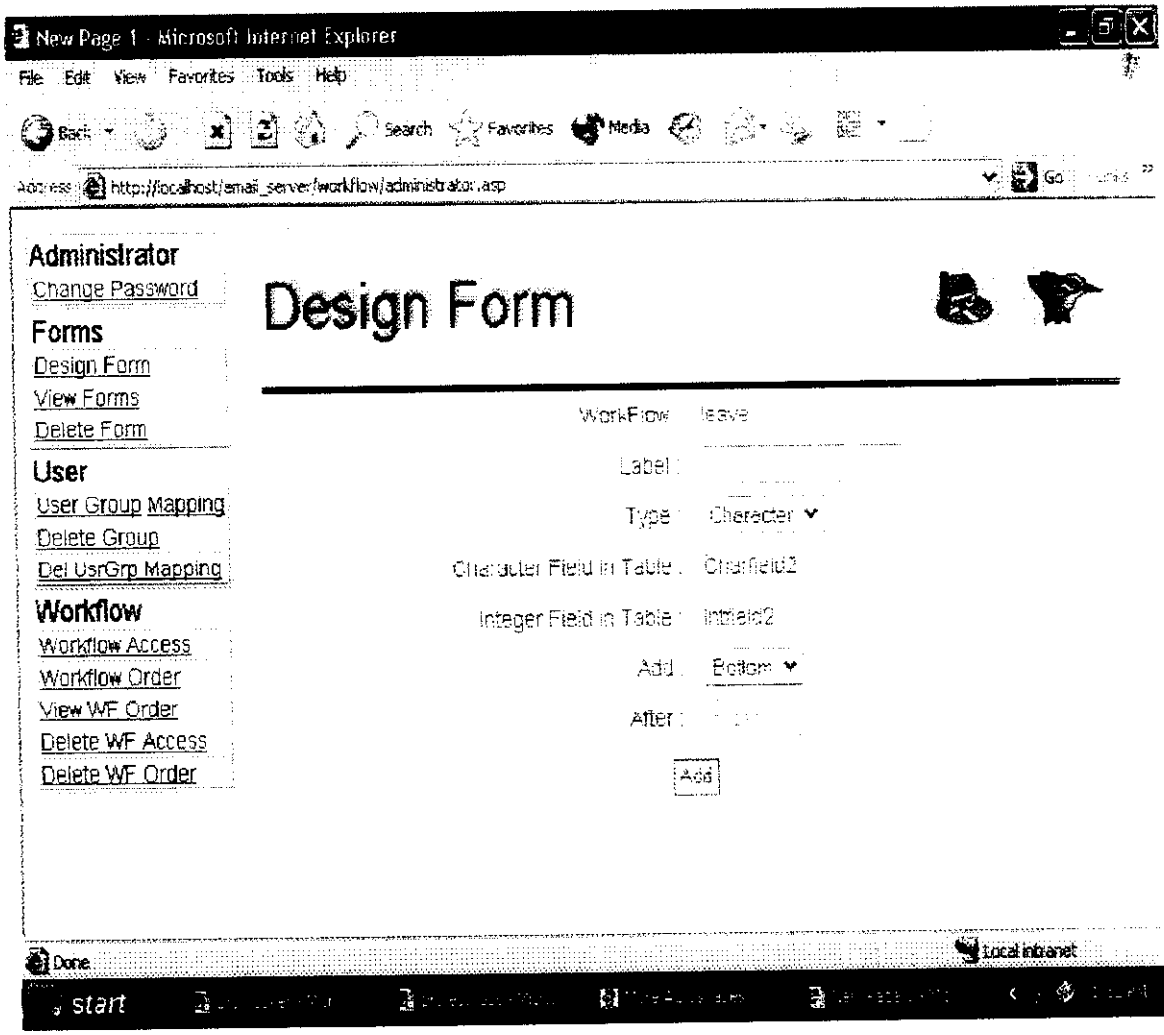
Includes chat_top, chat_mid, chat_bottom.



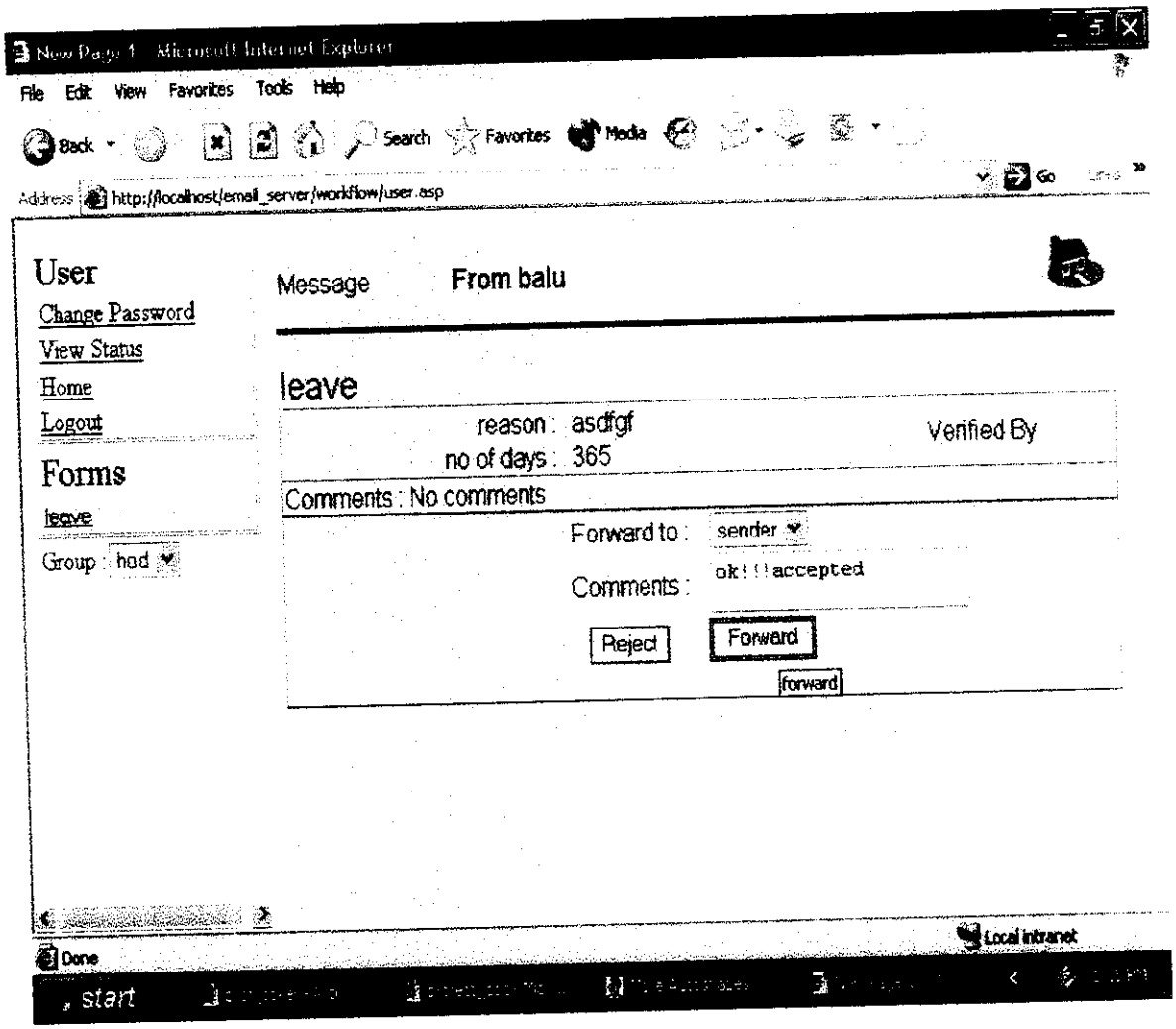
Event remainder



Work Flow-Administrator Login



Work Flow-forms



Coding

Verify. Asp

```
<%
'Form to check the status of user login
Option Explicit
dim dbname
dim myname, mypassword
Dim conntemp, cnpath, sqltemp, rstemp, rs, rs1
dbname="email_server.mdb"
myname=request.form("txtUserid")
mypassword=request.form("txtPassword")
Set conntemp=server.createobject("adodb.connection")
cnpath="DBQ=" & server.mappath(dbname)
conntemp.Open "DRIVER={Microsoft Access Driver (*.mdb)}; " & cnpath
sqltemp="select * from members where username='"
sqltemp=sqltemp & myname & "'"
Set rstemp=conntemp.execute(SQLTemp)

If rstemp.eof then

   rstemp.close
   conntemp.close
   Set rstemp=nothing
   Set conntemp=nothing
    Session ("Message") = "Invalid Username. Please, try again."
   response.redirect "Default.asp"
End if

If rstemp("password")=mypassword then

    set rs=server.CreateObject("adodb.recordset")
    set rs=conntemp.Execute("select status from members where
username='"&myname&"'"&"and password='"&mypassword&"'")

    if rs("status")=1 then
     rstemp.close
    conntemp.close
    set rstemp=nothing
    set conntemp=nothing
    Session("message")="User Already Logged On"
    Response.Redirect"Default.asp"

    else
    set rs1=server.CreateObject("adodb.recordset")
    set rs1=conntemp.Execute("update members set status = '1' where
username='"&myname&"'"&"and password='"&mypassword&"'")
    rstemp.close
    conntemp.close
    set rstemp=nothing
    set conntemp=nothing
```

```
        If Request.Form("RememberMe") = "True" Then
        Response.Cookies("Username") = request("txtUserid")
        Response.Cookies("Password") = request("txtPassword")
        Response.Cookies("Username").Expires = DateAdd("m", 1, Now())
        Response.Cookies("Password").Expires = DateAdd("m", 1, Now())
        end if


     Session("mess")= myname
     response.redirect "main.asp"
    end if
else
     rstemp.close
    conntemp.close
    set rstemp=nothing
    set conntemp=nothing

     Session("Message") = "Invalid password. Please, try again."
          response.redirect "Default.asp"
end if
%>
```

Message. Asp

```
<%
'to store the message in the database
option explicit
Sub DrawPage
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>New Page 1</title>
</head>

<body bgcolor="#FFFFFF">
<form name="messa" method=post action="delete.asp">
<%
     dim conn,DSNtemp,rs,a,rs1,SQLstmt
     set conn=server.CreateObject("adodb.connection")
     set rs=Server.CreateObject("adodb.recordset")
    set rs1=server.CreateObject("adodb.recordset")
    DSNtemp = "DRIVER={Microsoft Access Driver (*.mdb)}; " & "DBQ=" &
 server.mappath("db1.mdb")
    conn.Open DSNtemp
    set rs=conn.Execute("select * from receive")
    a = CInt(Request.QueryString("id"))
    Session("id")=a
```

```
  if not rs.EOF then
    rs.MoveFirst
    do until  rs.Fields("id") = a
     rs.MoveNext
    loop
  end if
  set rs1=conn.Execute("update receive set status = 1 where id = " & a)
     %>
<P><b>FROM    :</b><% =
rs("from1")%>         <a
href="../mailsr/inbox.asp"><b>Back to Inbox</b></a>
<p><b>SUBJECT :</b> <% = rs("subject")%></p>
<p><b>CC      :</b> <% = rs("cc")%></p>
<P><b>DATE    :</b> <% = rs("res_dat")%></P>
<P><b>MESSAGE :</b>     </P>
<P>           &n
bsp;      
<TEXTAREA name=S1 rows=7 cols=68 tabindex="1" > <% =
rs("message")%></TEXTAREA>
 <input type="submit" value="Delete" name="Delete">
<P> </P>

<%
SQLstmt = "INSERT INTO receive (from1,toe,subject,message,res_dat)"
       SQLstmt = SQLstmt & " VALUES ("
       SQLstmt = SQLstmt & "'" & mess & "',"
    SQLstmt = SQLstmt & "'" & rs("from1") & "',"
    SQLstmt = SQLstmt & "'" & rs("subject") & " ',"
       SQLstmt = SQLstmt & "' Your mailwith message "&rs("message")&" is viewed
',"
       SQLstmt = SQLstmt & "'" & now() & "'"
      SQLstmt = SQLstmt & ")"
      Set rs = conn.execute(SQLstmt)
       conn.Close %>

</form>
</body>
</html>
<%
  End Sub
  dim mess
  mess=session("mess")
    call DrawPage
  %>
```

Inbox. Asp

```
<%
option explicit
Sub DrawPage()
```

```
%>
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-
1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>New Page 1</title>
</head>

<body>
<form name="inbox" method="post" >
<p>  <img border="0"
src="../workflow/images/mailbox.gif">   Inbox of
<%=mess%>          
<a href="../main.ASP"><img border="0"
src="../workflow/images/home.gif"></a>
<p> 

<div align="center">
  <center>
  <table border="1" width="100%">
    <tr>
      <th width="25%" align="center" bgcolor="#3A8B00"><font
color="#FFFFFF">STATUS</font></th>
      <th width="25%" align="center" bgcolor="#3A8B00"><font
color="#FFFFFF">FROM</font></th>
      <th width="25%" align="center" bgcolor="#3A8B00"><font
color="#FFFFFF">DATE</font></th>
      <th width="25%" align="center" bgcolor="#3A8B00"><font
color="#FFFFFF">SUBJECT</font></th>
    </tr>
    <%
     dim conn,rs1,DSNtemp
     set conn = Server.CreateObject("adodb.connection")
     set rs1 = server.CreateObject("adodb.recordset")
     DSNtemp = "DRIVER={Microsoft Access Driver (*.mdb)}; " & "DBQ=" &
server.mappath("db1.mdb")
     conn.Open DSNtemp
     set rs1 = conn.Execute("select * from receive where toe = "&"'"&
mess &"'")
     do until rs1.EOF

     %>
       <tr>
         <td width="25%" align="center" bgcolor="#F8FFEC">
            <font color="#3A8B00"> 
         <% If rs1("status")= 0 then %>
           <img src ="../mailsr/hlp.bmp">
           <% else %>
           <img src="../mailsr/openbook.gif">
           <% end if%>
            </font>
           </td>
          <td width="25%" align="center" bgcolor="#F8FFEC"><font
color="#3A8B00"><%=rs1("from1")%></font></td>
```

```
        <td width="25%" align="center" bgcolor="#F8FFEC"><font
color="#3A8B00"><%=rs1("res_dat")%></font></td>
        <td width="25%" align="center" bgcolor="#F8FFEC">
        <a href="../mailsr/message.asp?ID= <% =rs1.Fields("ID")%> ">
<font color="#3A8B00"> <%=rs1("subject")%></font></a>
       </td>
      </tr>
      <%
        rs1.MoveNext
        loop
        %>
  </table>
  </center>
</div>

<% conn.Close %>
<p>
 
</p>
 
</form>
</body>
</html>

<%
End Sub
dim mess
mess=session("mess")
call DrawPage()
%>
```

Chat_top. Asp

```
<%
option explicit
sub DrawPage
%>

<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">   <script language="JavaScript">
<!--
function MM_openBrWindow(theURL,winName,features) {
  window.open(theURL,winName,features);
}
//-->
</script>
<base target="_self">
</head>
<body bgcolor="#5C7AA1" text="#FF6699" leftmargin="0" topmargin="0"
marginwidth="0" marginheight="0"
onUnload="MM_openBrWindow('logout.asp','','width=1,height=1')">
<p><font size="7"><b>Welcome to the chat ROOM!</b></font></p>

<%=send%>
```

```
</body>
</html>

<%
end sub
dim send
send=session("send")
session("send")=""
call DrawPage
%>
```

Chat_mid. asp

```
<%
session("send")=Request.Form("msg")
If Session("Refresh") = "" Then Session("Refresh") = "2000"
%>
<script language="JavaScript">

   function update()
{ document.location.href="chat_mid.asp"

}
   function toBottom(){
   if(document.all){
                  window.scroll(0, document.body.clientHeight);
                  }
   else{
      window.scroll(0,20000)
       }
                  }
   setTimeout("toBottom()", 1)
   setTimeout("update()",<% response.write Session("Refresh") %>);
</script>
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
<base target="_self">
</head>

<body bgcolor="#006699" text="#FF6699" topmargin="0" marginheight="0">
<table width="600" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td> <font face="verdana" size="1">User online:
    <%
    Set Con = Server.CreateObject("ADODB.Connection")
    Set rsQuery = Server.CreateObject("ADODB.Recordset")
    Path = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="
    Path = Path & Server.MapPath("db/chat.mdb") & ";Mode=ReadWrite"
    Con.Open Path

    Set rsQuery = Con.Execute("SELECT * FROM logdb")
    While Not rsQuery.EOF
    Response.Write rsQuery("Benutzer") & ", "
    rsQuery.MoveNext
```

```
   Wend
   %>
   </font> </td>
  </tr>
</table>
<table width="600" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td>
   <font face=verdana size=2>
      <%
t = 1
Set rsQuery =  Con.Execute("SELECT * FROM msgdb WHERE An='@all' OR
An='" & Session("Name") & "' ORDER BY ID ASc")
Dim Text(50)
Do Until rsQuery.EOF
   If Instr(DateDiff("s", Session("Start"), rsQuery("Zeit")), "-") = 0
Then
   If rsQuery("Von") <> Session("Ignore") Then
   Text(t) =  rsQuery("Nachricht") & "<br>"
   t = t + 1
   End If
   End If
   rsQuery.MoveNext
Loop
For i = 1 to t
   If i <> t Then Response.Write Text(i)
   If i = t Then Response.Write Text(i) & "<a name=new></a>"

Next

%>
    </font>
    </td>
  </tr>
</table>
</body>
</html>
```

Add_event. Asp

```
<%@ LANGUAGE=VBSCRIPT %>
<%Option Explicit%>

<%
Dim DB_CONNECTIONSTRING
Dim objRecordset
Dim Added
%>
   <!--#include file="adovbs.inc"-->
<%
DB_CONNECTIONSTRING = "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.Mappath("users.mdb") & ";"

Set objRecordset = Server.CreateObject("ADODB.Recordset")
objRecordset.Open "calendar", DB_CONNECTIONSTRING, adOpenStatic,
adLockPessimistic, adCmdTable
```

```
If Request.Form("btnAdd") = "Add Event" Then
        objRecordset.AddNew


   '-- Add records to database from form --
         objRecordset.Fields("Subject") = Request.Form("txtSubject")
         objRecordset.Fields("Message") = Request.Form("Message")
         objRecordset.Fields("Day") = Request.Form("selDay")
         objRecordset.Fields("Month") = Request.Form("selMonth")
         objRecordset.Fields("Year") = Request.Form("selYear")
         objRecordset.Fields("AddedBy") = Request.Form("txtAddedBy")
         objRecordset.Fields("DateAdded") = Now()

       objRecordset.Update
       Added = "True"

End If

objRecordset.Close
Set objRecordset = Nothing

If Added = "True" Then
   Response.Redirect("calendar.asp")
End If
%>

<html>
<center>

<form method="post" action="add_event.asp">

<table border="0" cellpadding="0" cellspacing="0">
 <tr>
  <td>Day:

    <select name="selDay">

    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
    <option value="7">7</option>
    <option value="8">8</option>
    <option value="9">9</option>
    <option value="10">10</option>
    <option value="11">11</option>
    <option value="12">12</option>
    <option value="13">13</option>
    <option value="14">14</option>
    <option value="15">15</option>
    <option value="16">16</option>
    <option value="17">17</option>
    <option value="18">18</option>
    <option value="19">19</option>
    <option value="20">20</option>
```

```
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<Option value="27">27</option>
<Option value="28">28</option>
<Option value="29">29</option>
<Option value="30">30</option>
<Option value="31">31</option>

</select></td>

<td>Month:
 <Select name="selMonth">

 <OPTION VALUE="1">January</option>
 <OPTION VALUE="2">February</option>
 <OPTION VALUE="3">March</option>
 <OPTION VALUE="4">April</option>
 <OPTION VALUE="5">May</option>
 <OPTION VALUE="6">June</option>
 <OPTION VALUE="7">July</option>
 <OPTION VALUE="8">August</option>
 <OPTION VALUE="9">September</option>
 <OPTION VALUE="10">October</option>
 <OPTION VALUE="11">November</option>
 <OPTION VALUE="12">December</option>

 </select></td>

 <td>Year:
  <Select name="selYear">

  <Option value="1999">1999</option>
  <Option value="2000">2000</option>
  <Option value="2001">2001</option>
  <Option value="2002">2002</option>
  <Option value="2003">2003</option>
  <Option value="2004">2004</option>

  </select></td>
</tr>
<tr>
 <td colspan="3">  </td>
</tr>
<tr>
 <td colspan="3" align="center">Subject: <input type="text"
name="txtSubject" size="35"></td>
</tr>
<tr>
 <td colspan="3"> </td>
</tr>
<tr>
 <td colspan="3" align="center"><textarea name="Message" cols="40"
rows="10"></textarea></td>
```

```
</tr>
<tr>
 <td colspan="3"> </td>
</tr>
<tr>
 <td colspan="3" align="center">Added By: <input type="text"
name="txtAddedBy" size="35" value="<%= Session("FirstName") & " " &
Session("LastName")%>"></td>
 </tr>
</table>
<p>
<input type="submit" name="btnAdd" value="Add Event">  <input
type="Reset" name="btnReset" value="Clear">

</form>
</center>
</body>
</html>
```

Edit_event. Asp

```
<%
session("send")=Request.Form("msg")
If Session("Refresh") = "" Then Session("Refresh") = "2000"
%>
<script language="JavaScript">

   function update(){
         document.location.href="chat_mid.asp"

               }

   function toBottom(){
   if(document.all){
                  window.scroll(0, document.body.clientHeight);
               }
   else{
      window.scroll(0,20000)
      }
                     }
   setTimeout("toBottom()", 1)
   setTimeout("update()",<% response.write Session("Refresh") %>);
 </script>
 <html>
 <head>
 <title>Untitled Document</title>
 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
 1">
 <base target="_self">
 </head>
 <body bgcolor="#006699" text="#FF6699" topmargin="0" marginheight="0">
 <table width="600" border="0" cellpadding="0" cellspacing="0">
   <tr>
     <td> <font face="verdana" size="1">User online:
    <%
    Set Con = Server.CreateObject("ADODB.Connection")
```

```
   Set rsQuery = Server.CreateObject("ADODB.Recordset")
   Path = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="
   Path = Path & Server.MapPath("db/chat.mdb") & ";Mode=ReadWrite"
   Con.Open Path

   Set rsQuery = Con.Execute("SELECT * FROM logdb")
   While Not rsQuery.EOF
   Response.Write rsQuery("Benutzer") & ", "
   rsQuery.MoveNext
   Wend
   %>
   </font> </td>
  </tr>
</table>
<table width="600" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td>
   <font face=verdana size=2>
      <%
t = 1
Set rsQuery =  Con.Execute("SELECT * FROM msgdb WHERE An='@all' OR
An='" & Session("Name") & "' ORDER BY ID ASc")
Dim Text(50)
Do Until rsQuery.EOF
   If Instr(DateDiff("s", Session("Start"), rsQuery("Zeit")), "-") = 0
Then
   If rsQuery("Von") <> Session("Ignore") Then
   Text(t) =  rsQuery("Nachricht") & "<br>"
   t = t + 1
   End If
   End If
   rsQuery.MoveNext
Loop
For i = 1 to t
   If i <> t Then Response.Write Text(i)
   If i = t Then Response.Write Text(i) & "<a name=new></a>"

Next
%>
   </font>
   </td>
  </tr>
</table>
</body>
</html>
```

Interform_s.asp

```
interform.asp
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft FrontPage 4.0">
<TITLE></TITLE>
</HEAD>
<BODY>
```

```
<%
set conn=server.CreateObject("adodb.connection")
set res1=server.CreateObject("adodb.recordset")
set res2=server.CreateObject("adodb.recordset")
cnpath="DBQ=" & server.mappath("workflow.mdb")
conn.Open "DRIVER={Microsoft Access Driver (*.mdb)}; " & cnpath
set res1=conn.Execute("select workflowname,wid from workflow where
wid=(select max(wid) from workflow)")
if not res1.EOF then
   if Request.Form("sel")="new" then
      set res2=conn.Execute("select workflowname,wid from workflow
where workflowname='" & Request.Form("new") & "'")
      if res2.EOF then
         conn.Execute("insert into workflow(workflowname,wid) values('"
& Request.Form("new") & "'," & res1("wid")+1 & ")")
         conn.Execute("insert into
transact(processid,userid,wid,charfield1,charfield2,charfield3,charfiel
d4,charfield5,charfield6,charfield7,charfield8,charfield9,charfield10,"
& _
"intfield1,intfield2,intfield3,intfield4,intfield5,intfield6,intfield7,
intfield8,intfield9,intfield10,charfield,intfield) values(0,0," &
res1("wid")+1 & ",'nu','nu','nu','nu','nu','nu','nu','nu','nu','nu'," &
_
                    "0,0,0,0,0,0,0,0,0,0,10,10)")
         session("wf")=Request.Form("new")
      else
         conn.Close
         Response.Redirect "../workflow/message/dupworkflow.htm"
      end if
   else
      set res1=conn.Execute("select workflowname from workflow where
wid=" & Request.Form("exist") & "")
      session("wf")=res1("workflowname")
   end if
else
    conn.Execute("insert into workflow(workflowname,wid) values('" &
Request.Form("new") & "'," & 1 & ")")
    conn.Execute("insert into
transact(processid,userid,wid,charfield1,charfield2,charfield3,charfiel
d4,charfield5,charfield6,charfield7,charfield8,charfield9,charfield10,"
& _
"intfield1,intfield2,intfield3,intfield4,intfield5,intfield6,intfield7,
intfield8,intfield9,intfield10,charfield,intfield)
values(0,0,1,'nu','nu','nu','nu','nu','nu','nu','nu','nu','nu'," & _
               "0,0,0,0,0,0,0,0,0,0,10,10)")

   session("wf")=Request.Form("new")
end if
conn.Close
Response.Redirect "../workflow/form.asp"
%>
</BODY>
</HTML>
```

Respond_s.asp

```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft FrontPage 4.0">
</HEAD>
<BODY>
<%
set conn=server.CreateObject("adodb.connection")
set res1=server.CreateObject("adodb.recordset")
set res2=server.CreateObject("adodb.recordset")
cnpath="DBQ=" & server.mappath("workflow.mdb")
conn.Open "DRIVER={Microsoft Access Driver (*.mdb)}; " & cnpath
set res1=conn.Execute("select * from history where processid=" &
trim(Request.Form("pid")))
set res2=conn.Execute("select dgid from wforder where dgid=(select
min(dgid) from wforder where wid=" & res1("wid") & " and ugid=(select
groupid from login where userid=" & res1("userid") & "))")
if Request.Form("but")="forward" then
   if Request.Form("D1")="sender" then
      conn.Execute("update history set
returnstatus=1,viewstatus=0,duserid=" & res1("userid") &
",forwarded=1,fdate_time='" & now() & "' where processid=" &
trim(Request.Form("pid")) & " and ouserid=" & session("userid") & "")
      conn.Execute("insert into
comment(processid,verifiedby,comment,vdate) values(" &
res1("processid") & ",'" & session("user") & "','" & Request.Form("S1")
& "','" & now() & "')")
      conn.Close
      Response.Redirect "../workflow/message/success.htm"
   else
         conn.Execute("update history set returnstatus=0,duserid=" &
   Request.form("D1") & ",forwarded=1,fdate_time='" & now() & "' where
   processid=" & trim(Request.Form("pid")) & " and ouserid=" &
   session("userid") & "")
   conn.Execute("insert into
history(processid,userid,ouserid,wid,viewstatus,odate_time,forwarded,st
atus) values(" & res1("processid") & "," & res1("userid") & "," &
Request.Form("D1") & "," & res1("wid") & ",0,'" & now() & "',0,0)")
conn.Execute("insert into comment(processid,verifiedby,comment,vdate)
values(" & res1("processid") & ",'" & session("user") & "','" &
Request.Form("S1") & "','" & now() & "')")
conn.Close
Response.Redirect "../workflow/message/success.htm"
end if
elseif Request.Form("but")="delete" then
conn.Execute("update history set status=1,forwarded=1 where processid="
& Request.Form("pid") & " and userid=duserid")
conn.Execute("delete from comment where processid=" &
Request.Form("pid") & "")
   set r1=server.CreateObject("adodb.recordset")
   set r1=conn.Execute("select processid from history where
returnstatus=1 and status=1 and userid=" & session("userid") & "")
   do until r1.EOF
```

```
   conn.Execute("insert into
backuphistory(processid,userid,ouserid,duserid,wid,viewstatus,odate_tim
e,vdate_time,fdate_time,forwarded,status,returnstatus)" & _
 " select
processid,userid,ouserid,duserid,wid,viewstatus,odate_time,vdate_time,f
date_time,forwarded,status,returnstatus from history where processid="
& r1("processid") & "")
conn.Execute("delete from history where processid=" & r1("processid") &
"")
conn.Execute("insert into
backuptransact(processid,userid,wid,intfield1,intfield2,intfield3,intfi
eld4,intfield5,intfield6,intfield7,intfield8,intfield9,intfield10,charf
ield1,charfield2,charfield3,charfield4,charfield5,charfield6,charfield7
,charfield8,charfield9,charfield10,intfield,charfield)" & _
 " select
processid,userid,wid,intfield1,intfield2,intfield3,intfield4,intfield5,
intfield6,intfield7,intfield8,intfield9,intfield10,charfield1,charfield
2,charfield3,charfield4,charfield5,charfield6,charfield7,charfield8,cha
rfield9,charfield10,intfield,charfield from transact where processid="
& r1("processid") & "")
   conn.Execute("delete from transact where processid=" &
r1("processid") & "")
   r1.MoveNext
   loop

   conn.close
   Response.Redirect "../workflow/message/messagedeleted.htm"
else
   conn.Execute("update history set
returnstatus=1,viewstatus=0,duserid=" & res1("userid") &
",forwarded=1,fdate_time='" & now() & "' where processid=" &
trim(Request.Form("pid")) & " and ouserid=" & session("userid") & "")
   if Request.Form("S1")="" then
   conn.Execute("insert into
comment(processid,verifiedby,comment,vdate) values(" &
res1("processid") & ",'" & session("user") & "','Rejected','" & now() &
"')")
else
   conn.Execute("insert into
comment(processid,verifiedby,comment,vdate) values(" &
res1("processid") & ",'" & session("user") & "','" & Request.Form("S1")
& "','" & now() & "')")
   end if
   conn.Close
   Response.Redirect "../workflow/message/success.htm"
end if
conn.Close
%>
</BODY>
</HTML>
```

Formselect_s.asp

```
<%@ Language=VBScript %>
<HTML>
```

```
<HEAD>
<META NAME="GENERATOR" Content="Microsoft FrontPage 4.0">
</HEAD>
<BODY>
<script language=VBScript runat=server>
private function f1()
   if Request.Form("type")="char" then
      conn.Execute("update transact set charfield=charfield-1 where
wid=" & Request.Form("wid") & "")
   else
      conn.Execute("update transact set intfield=intfield-1 where wid="
& Request.Form("wid") & "")
   end if
end function
</script>
<%
set conn=server.CreateObject("adodb.connection")
set res1=server.CreateObject("adodb.recordset")
set res2=server.CreateObject("adodb.recordset")
set res3=server.CreateObject("adodb.recordset")
set res4=server.CreateObject("adodb.recordset")
set res5=server.CreateObject("adodb.recordset")
cnpath="DBQ=" & server.mappath("workflow.mdb")
conn.Open "DRIVER={Microsoft Access Driver (*.mdb)}; " & cnpath
set res3=conn.Execute("select * from form where wid=" &
Request.Form("wid") & " and label='" & Request.Form("label") & "'")
if res3.EOF then
   if Request.Form("type")="char" then
      temp1=Request.Form("fchar")
   else
      temp1=Request.Form("fint")
   end if
   set res5=conn.Execute("select * from form where wid=" &
Request.Form("wid") )
   if res5.EOF then
      conn.Execute("insert into form(wid,label,oposition,fieldintable)
values(" & Request.Form("wid") & ",'" & Request.Form("label") & "',1,'"
& temp1 & "')")
      call f1
      conn.Close
      Response.Redirect "../workflow/form.asp"
   else
      if Request.Form("addat")="bottom" then
         set res2=conn.Execute("select oposition from form where
oposition=(select max(oposition) from form where wid=" &
Request.Form("wid") & ")")
         if temp1<>"no" then
            conn.Execute("insert into
form(wid,label,oposition,fieldintable) values(" & Request.Form("wid") &
",'" & Request.Form("label") & "'," & res2("oposition")+1 & ",'" &
temp1 & "')")
            call f1
         end if
         conn.Close
         Response.Redirect "../workflow/form.asp"
      elseif Request.Form("addat")="top" then
         if temp1<>"no" then
```