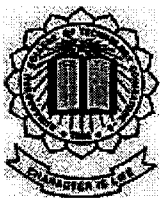
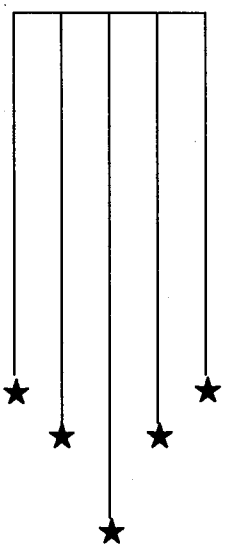
MAIL SERVER MANAGEMENT AND SECURITY USING KEYSTROKE DYNAMICS

PROJECT WORK





Submitted by

Ilango .S
Rajesh .K
Ramkumar .P

Under the guidance of

Mrs. R. K .Kavitha M.C.A
Computer Technology Department

In partial fulfillment of the requirements
For the award of the degree of

BACHELOR OF SCIENCE
(Applied Science -Computer Technology)

of the BHARATHIAR UNIVERSITY, Coimbatore.

DEPARTMENT OF COMPUTER TECHNOLOGY

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641 006.

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE:641 006

DEPARTMENT OF COMPUTER TECHNOLOGY

CERTIFICATE

THIS IS TO CERTIFY THAT THIS PROJECT ENTITLED

MAIL SERVER MANAGEMENT AND SECURITY USING KEYSTROKE DYNAMICS

HAS BEEN SUBMITTED BY MR. S. RANGO, K. RAJESH, P. RAMKUMAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE APPLIED SCIENCE COMPUTER TECHNOLOGY OF BHARATHIAR UNIVERSITY, COIMBATORE:641 046 DURING THE ACADEMIC YEAR 2002-2003.

R.K. Kavitha 29/3/03

(GUIDE)

(HEAD OF DEPARTMENT)

CERTIFIED THAT THE CANDIDATE WAS EXAMINED BY US IN THE PROJECT WORK VIVA-VOCE EXAMINATION HELD ON 25-03-2003

UNIVERSITY REGISTER NUMBER 0028Q0128, 0028Q0131, 0028Q0153

(INTERNAL EXAMINER)

(EXTERNAL EXAMINER)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I wish to express my sincere and heartfelt gratitude to Dr. K. K. Padmanabhan, B. Sc (Engg), M. Tech., Ph.D., our esteemed principal, Kumaraguru College of Technology for giving us the needed encouragement for this project.

With profound sense of gratitude and regards ,we acknowledge with great pleasure and guidance and support extended by Dr. Sundaram, Head of the Department of Science and Humanities, for his constructive criticisms and persistent encouragement.

We express our deep sense of respectful gratitude to our guide Mrs.K. Kavitha, Department Of Computer Technology for her valuable and enthusiastic guidance, keen suggestions, innovative ideas, inspiration, discussion, helpful criticisms and kind encouragement in all the phases of this project work. I also thank our class advisor Mr. S. Hameed Ibrahim M.C.A and our course coordinator for their valuable guidance.

Finally I express my thanks to all Staff members and Lab Technicians in the Department who helped me for the successful completion of the project. I also express my deep sense of gratitude to my parents, friends and all others who had been directly or indirectly involved with this project, for their invaluable help and consideration towards me.

SYNOPSIS

The software entitled as **'Mail Server Management & Security Using Keystroke Dynamics'** is developed using the front-end tool Visual Basic 6.0 and MS Access as a back-end tool.

The Mail Server Management & Security Using Keystroke Dynamics deals with providing e-mail access to the clients over a LAN network without allowing them to browse the net. It also manages the appointments of the individuals working in the company.

The Mail Server Management & Security Using Keystroke Dynamics is a network based system. The use of this project allows saving the time conception and reduces the high utilization of the net and the clients are blocked from using the net for their personal use.

The Mail Server Management & Security Using Keystroke Dynamics provides security using a new concept called as Keystroke Dynamics.

Source code clarity is enhanced by structured coding technique, good coding style and appropriate supporting good internal comments. Database design has been done in order to handle information as an integrated one.

Code design has been designed by considering the factors like response time, volume of data used, hardware constraints, cost factors etc. Data flow diagrams, sample coding and sample screen designs are attached.

The testing phase is an important part of software development. It is the process of finding errors and missing operation and also a complete verification of whether all the objectives are met and the user requirements are satisfied. The software testing is carried out in several steps.

CONTENTS

PROJECT OVERVIEW:

PROJECT OVERVIEW:

PROVIDING E-MAIL ACCESS TO ALL THE EMPLOYEES IS A MAJOR NECESSITY IN MOST ORGANIZATIONS. ACHIEVING SUCH A FUNCTIONALITY USING PROXY SERVERS INVOLVES VIRUS INTRUSION AND SERVER'S ACCESS SYNCHRONIZATION BETWEEN EMPLOYEES BECOMES HARD TO ACHIEVE. ALSO THE PROXY CAN BE MISUSED BY WORKERS FOR NET- SURFING.

AN ALTERNATIVE SOLUTION HAS BEEN ATTEMPTED IN THIS PROJECT TO USE A COMMON MAIL ID AMONG THE WORKERS FOR BOTH SENDING AND RECEIVING MAILS. THE APPLICATION INSTALLED IN EVERY CLIENT MACHINE GETS THE MAIL FROM THE EMPLOYEES, CONNECTS TO THE INTERNET WITH THE HELP OF THE SYSTEM ADMINISTRATOR & SENDS THE MAIL THROUGH THE COMMON MAIL ID BY DYNAMICALLY CHANGING THE DISPLAY NAMES.

THE APPLICATION ALSO DOWNLOADS ALL THE MAILS THAT ARRIVE AT THE COMMON INBOX AND DISPATCHES IT TO THE RESPECTIVE LOCAL BOXES.

THE SECURITY ISSUE IS IMPLEMENTED BY USING A METHOD CALLED KEYSTROKE DYNAMICS, WHICH RECOGNIZES THE KEYSTROKE TYPING PATTERNS OF INDIVIDUALS AND PROVIDES AUTHENTICATION. THUS THIS METHOD ELIMINATES THE CURRENT DRAWBACK OF EXISTING SYSTEM, IN WHICH PASSWORDS CAN BE GUESSED AND HACKED BY MEANS OF BRUTE-FORCE ATTACKS EVEN IF ENCRYPTION IS USED.

ALSO, THE INTERNAL WORK FLOW IN THE ORGANIZATION IS AUTOMATED BY MAINTAINING APPOINTMENTS OF INDIVIDUALS AND SENDING REMAINDERS TO THEM. THE USER LEVEL APPLICATION ACCESS IS ALSO ACHIEVED SO THAT THE EMPLOYEES ARE BLOCKED FROM ACCESS TO SPECIFIED APPLICATION.

THUS THE PROJECT IS A COMPLETE UTILITY WHICH CAN BE INCORPORATED IN ORGANIZATIONS FOR ACHIEVING E-MAIL SOLUTIONS ALONG WITH MAINTAINING WORK SCHEDULES WITH A HIGH LEVEL OF SECURITY.

OBJECTIVE OF THE PROJECT

- Provide E-mail access to the employees working in an organization.
- The employees will not be allowed to surf the internet.
- Obtain user level access by blocking certain applications from the clients.
- The appointment of the individuals working in the organization can be maintained.
- Provides security using a new concept called as Key-stroke Dynamics

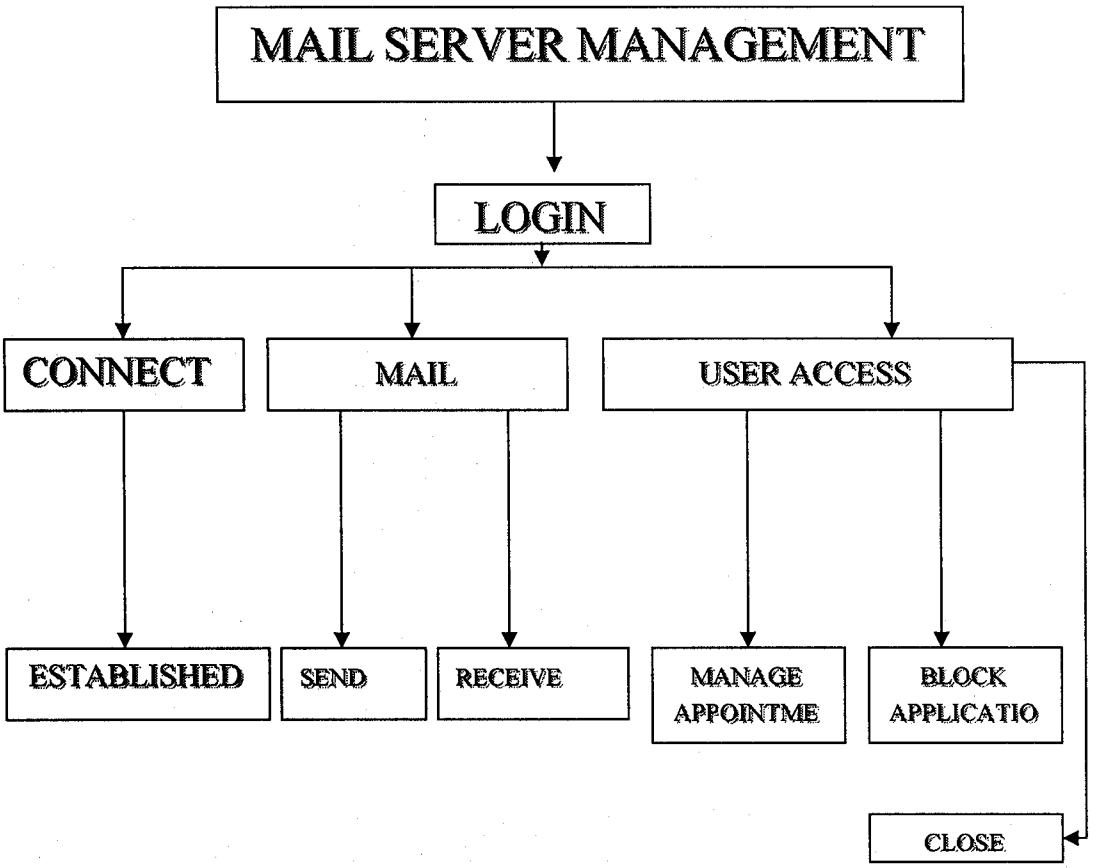
PRODUCT PERSPECTIVE

OUR PROJECT WILL BE USED IN A LAN NETWORK. MAIL SERVER MANAGEMENT CONSISTS OF TWO VISUAL BASIC PROJECTS.

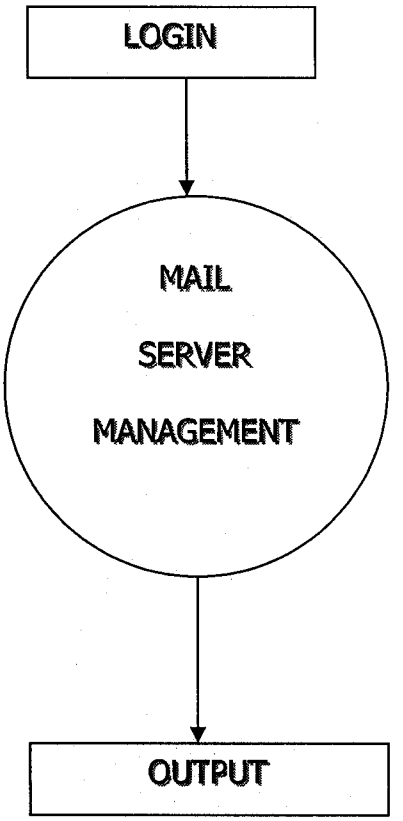
ONE VB PROJECT IS USED FOR MAINTAINING A SERVER. THE SERVER RECEIVES MAILS FROM THE CLIENT AND FORWARDS IT TO THE INTERNET. AT THE SAME TIME THE SERVER RECEIVES THE MAIL FROM THE INTERNET AND FORWARDS IT TO THE CLIENTS. THE SERVER ALSO CAN CHECK THE APPOINTMENT OF THE INDIVIDUALS.

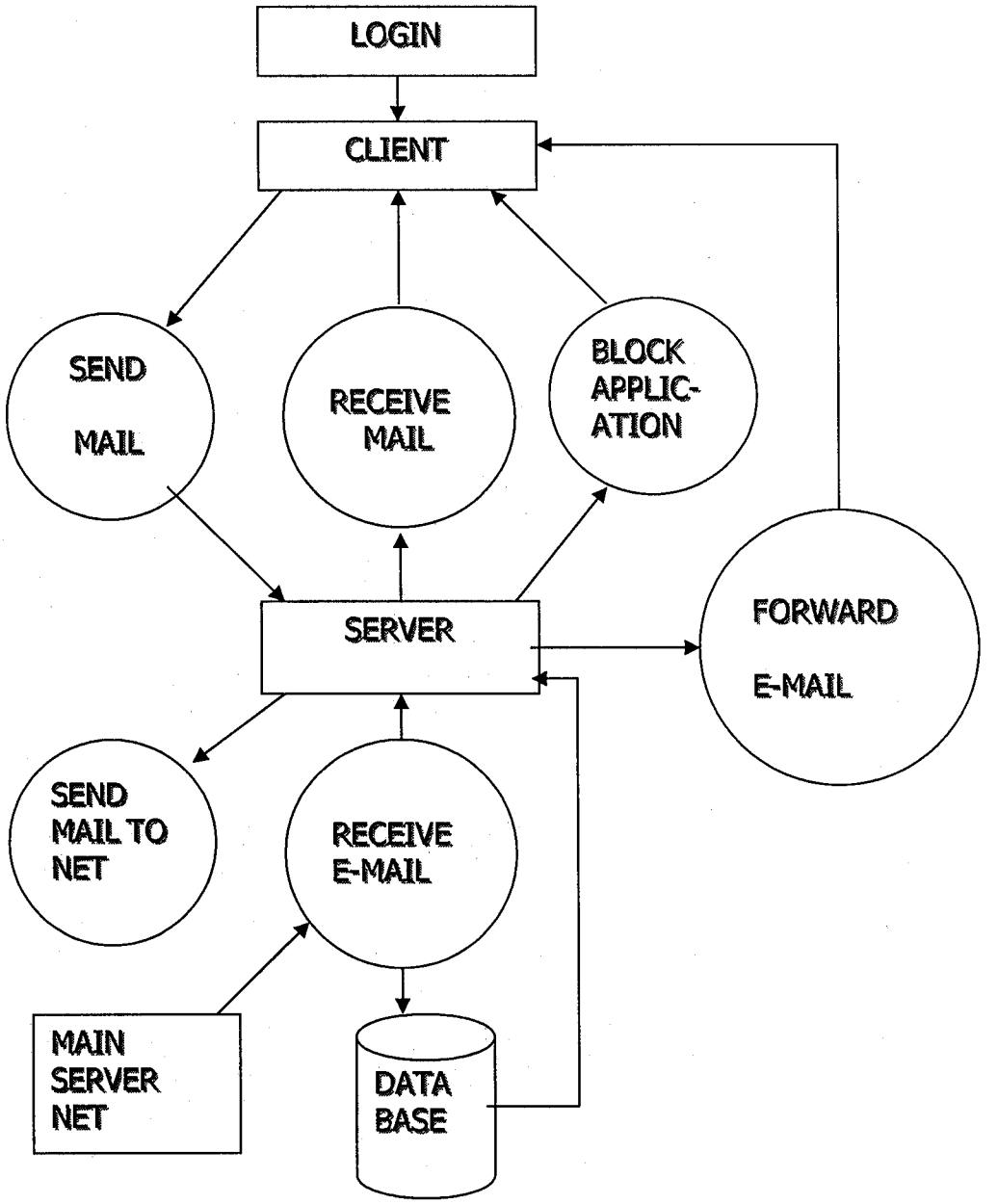
ANOTHER ONE IS USED AS A CLIENT. THE CLIENT WILL SEND MAIL TO THE SERVER AND RECEIVE MAILS FROM THE SERVER. A CLIENT WILL MANAGE THE APPOINTMENT OF THE INDIVIDUALS.

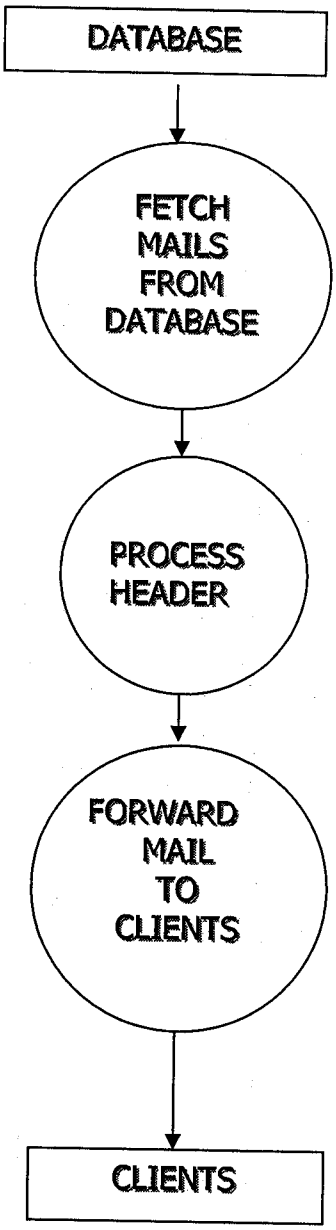
SYSTEM FLOW DIAGRAM



DATA FLOW DIAGRAM




LOGIN
CLIENT
SEND MAIL
RECEIVE MAIL
BLOCK APPLIC-ATION
SERVER
FORWARD E-MAIL
SEND MAIL TO NET
RECEIVE E-MAIL
MAIN SERVER NET
DATA BASE


DATABASE
FETCH MAILS FROM DATABASE
PROCESS HEADER
FORWARD MAIL TO CLIENTS
CLIENTS

PROGRAMMING ENVIRONMENT

SOFTWARE CONFIGURATION

Our Project is developed using:

Front-end : VB 6.0

Back-end : MS - Access

Operating System : WINDOWS 98, Windows 2000 server

VISUAL BASIC 6.0

Visual Basic is one of the components of Microsoft Visual Studio. Visual Basic evolved from BASIC (Beginner's All-purpose Symbolic Instruction Code). BASIC was developed in the mid- 1960s by Professors John Kemeny and Thomas Kurtz of Dartmouth College as a language for writing simple programs. BASIC's primary purpose was to help learn how to program.

The widespread use of BASIC with various types of computers (hardware platforms) led to many enhancements to the language. With the development of the Microsoft Windows graphical user interface (GUI) in the late 1980s and the early 1990s, the natural evaluation of BASIC was Visual Basic, which was created by Microsoft Corporation in 1991.

Until Visual Basic appeared, developing Microsoft Windows-based applications was a difficult and cumbersome process. Visual Basic greatly simplifies windows application development. Since 1991 six versions have been released, with Visual Basic 6- appeared in September 1998. The latest version is Visual Basic .NET which is rich in internet controls.

OVERVIEW OF VISUAL BASIC 6.0

Microsoft Visual Basic is the fastest and easiest way to create applications from Microsoft Windows. Whether you are an experienced professional or brand new to Windows Programming, Visual Basic provides you with a complete set of tools to simplify Rapid Application Development.

VISUAL BASIC IS OBJECT-ORIENTED, THAT IS, IT REVOLVES AROUND READYMADE OBJECTS. ONE OF THE MAIN IDEAS OF OBJECT - ORIENTED PROGRAMMING IS THAT ALL OF THE DATA AND PROCEDURES RELATED TO A PARTICULAR OBJECT ARE KEPT TOGETHER WITH THE OBJECT ITSELF. IN VISUAL BASIC AN OBJECT'S DATA ARE CALLED PROPERTIES, WHILE THE VARIOUS PROCEDURES THAT CAN OPERATE ON THE OBJECT ARE CALLED ITS METHODS. VISUAL BASIC IS THE SIMPLEST AND EASIEST-TO-USE PROGRAMMING LANGUAGE FOR THE WINDOWS ENVIRONMENT; IT HAS GROWN INTO WITH FAR-REACHING CAPABILITIES AND SOPHISTICATION. VISUAL BASIC IS EVENT DRIVEN.

THE LANGUAGE MAKES USE OF THE FEATURES OF MICROSOFT WINDOWS INCLUDING MULTIPLE DOCUMENT INTERFACE (MDI), OBJECT LINKING AND EMBEDDING (OLE), DYNAMIC DATA EXCHANGE (DDE), GRAPHICS, ACTIVEX CONTROLS, ETC. USING THIS WE CAN CREATE POWERFUL AND FULL-FEATURED APPLICATIONS. ADDING CUSTOM CONTROLS CAN EXTEND VISUAL BASIC AND CALLING PROCEDURES IN DYNAMIC LINK LIBRARIES (DLL) I.E. SPECIALLY CONSTRUCTED LIBRARIES CAN BE LOADED AND LINKED AT RUN TIME. MULTIPLE APPLICATIONS CAN SHARE DLLS, WHICH SAVES MEMORY AND DISK SPACE. DYNAMIC LINKING INCREASES PROGRAM MODULARITY BECAUSE YOU CAN COMPILE AND TEST DLLS SEPARATELY.

VISUAL BASIC HAS A LARGE NUMBER OF BUILT-IN OBJECTS THAT THE USER CAN USE WITH MAXIMUM FLEXIBILITY. THE ONLY TASK FOR THE PROGRAMMER IS TO INCORPORATE THE BUILT-IN OBJECTS TO HIS/HER PROGRAMS. THE CORE OF VISUAL BASIC PROGRAMMING IS A SET OF INDEPENDENT PIECES OF CODE THAT

ARE ACTIVATED BY AND SO RESPOND TO ONLY THE EVENTS THEY HAVE BEEN TOLD TO RECOGNIZE.

WHETHER YOUR GOAL IS TO CREATE A SMALL UTILITY FOR YOURSELF OR YOUR WORKGROUP, A LARGE ENTERPRISE-WIDE SYSTEM, OR EVEN DISTRIBUTED APPLICATIONS SPANNING THE GLOBE VIA INTERNET, VISUAL BASIC HAS THE TOOLS YOU NEED.

- ACTIVEX TECHNOLOGIES ALLOW YOU TO USE THE FUNCTIONALITY PROVIDED BY OTHER APPLICATIONS, SUCH AS MICROSOFT WORD PROCESSOR, MICROSOFT EXCEL SPREADSHEET, AND OTHER WINDOWS APPLICATIONS. YOU CAN EVEN AUTOMATE APPLICATIONS AND OBJECTS CREATED USING THE PROFESSIONAL OR ENTERPRISE EDITIONS OF VISUAL BASIC.

- INTERNET CAPABILITIES MAKE IT EASY TO PROVIDE ACCESS TO DOCUMENTS AND APPLICATIONS ACROSS THE INTERNET FROM WITHIN YOUR APPLICATION.

- YOUR FINISHED APPLICATION IS A TRUE.EXE FILE THAT USES A RUNTIME DYNAMIC LINK LIBRARY (DLL) THAT YOU CAN FREELY DISTRIBUTE.

THE DATA ENTRY FROM CREATED IN VISUAL BASIC CAN TAKE ITS DATA FROM DIFFERENT DBMS TABLES. HOWEVER, ONLY ONE DBMS CAN BE ACCESSED BY THE VISUAL BASIC FROM AT A GIVEN INSTANT OF TIME. HOWEVER, WHEN WRITING BACK TO DATA STORAGE STRUCTURE, VISUAL BASIC CAN BE INSTRUCTED

TO WRITE TO AN ENTIRELY DIFFERENT STRUCTURE WHEN COMPARED TO THE ONE FROM WHICH IS ACCESSED IT DATA.

FEATURES FOCUS OF VISUAL BASIC 6.0

- THE NEW PROJECT DIALOG BOX IS NOW VB 6.0
- CUSTOMIZABLE TOOL BARS THAT LET OPTIMIZE DESIGN ENVIRONMENT.
- THE PROPERTY DESCRIPTION FEATURE CAN SAVE TIME IN FINGERING OUT THE PURPOSE FOR OBJECT PROPERTIES.
- THE FORM LAYOUT WINDOWS.
- THE ABILITY TO ADD BREAK POINTS AND TO ALIGN CONTROLS USING A MENU.
- THE STARTUP POSITION PROPERTY, STYLE PROPERTY AND MOVABLE PROPERTY.
- NATIVE CODE COMPLETION IS ONE OF THE MOST EXITING NEW FEATURES.
- APPLICATION WIZARD EXPEDITES THE CREATION OF NEW PROJECT.
- WITH VB 6.0 WE CAN ABLE TO PUT PICTURE ON COMMAND BUTTON.

CREATING PROJECT IN VISUAL BASIC

VISUAL BASIC IS COMPLETELY DEVELOPMENT ENVIRONMENT. EVERY FUNCTIONS AND PROCEDURES WILL NEED TO WORK WITH VISUAL BASIC PROJECT IS ACCESS IS VISUAL BASIC. THE VISUAL BASIC ENVIRONMENT IS CALLED THE VISUAL BASIC INTEGRATED DESIGN ENVIRONMENT. TO WORK WITH VISUAL BASIC PROJECT YOU MUST FIRST START WITH VISUAL BASIC IDE. THE PROGRAM NAMED STANDARD. EXE IS SELECTED TO CREATE A NEW PROJECT, COMPLETE WITH ONE FORM TO WORK ON. A PROJECT IS WHAT YOU ARE ACTUALLY CREATING WHEN WORKING WITH IN THE COMPLETE COLLECTION OF FILES THAT MAKE UP THE

PROJECTS. A PROGRAM IS THE FINAL, COMPLIED VERSION OF THE PROJECT. ALL PHYSICAL "THINGS" WORK IN THE VISUAL BASIC PROJECT IS OBJECTS. THERE ARE MANY DIFFERENT TYPES OF OBJECTS. EVERY OBJECT HAS A DIFFERENT SET OF PROPERTIES.

PROPERTIES DEFINE THE CHARACTERISTICS OF THE OBJECTS. THE FORMAL THAT VISUAL BASIC CREATES WHEN A NEW PROJECT IS CREATED IS AN OBJECT. ALL ITEMS PUT ON FORMS SUCH AS LIST BOX AND COMMAND BUTTONS ARE OBJECTS. FORM OBJECTS IS A WINDOW OR A DIALOG BOX THAT MAKE UP PART OF AN APPLICATION INTERFACE. CONTROLS ARE OBJECTS ON THE FORMS THAT USERS INTERACT WITH. TOOLBOX IS USED TO SELECT CONTROLS TO PLACE ON THAT FORMS. COMMAND BUTTONS, TEXT BOX, OPTION BUTTON ARE SOME OF CONTROLS. WITH THE HELP OF THESE INTERFACES FOR OUR PROGRAMS ARE CREATED. THERE ARE ALSO MORE ADVANCED VISUAL BASIC CONTROLS SUCH AS DRIVE LIST BOX, DIR LIST BOX; FILE LIST BOX, TIMER CONTROL AND OLE CONTROL ARE AVAILABLE. WITH THE HELP OF THESE ACTIVE X CONTROLS WE CAN EASILY ALIGN CONTROLS ON A FORM AND SETTING TAB ORDER AND Z ORDERED. MORE OVER, CONTROL ARRAYS ADD FUNCTIONALLY AND REDUCE SYSTEM RESOURCES USED BY THE PROGRAM.

VISUAL BASIC AS THE DEVELOPMENT TOOL

OBJECT ORIENTED APPROACH

OBJECT ORIENTED PROGRAMMING IDENTIFIES THE WHOLE SYSTEM AS A COLLECTION OF FLEXIBLE ENTITIES CALLED OBJECTS INVOLVED. VISUAL BASIC IS OBJECT ORIENTED IN A BROAD SENSE.

IT HAS A LARGE NUMBER OF BUILD IN OBJECTS THAT THE USER CAN USE WITH MAXIMUM FLEXIBILITY. THE ONLY TASK FOR PROGRAMMER IS TO INCORPORATE THE BUILT-IN OBJECTS TO HIS PROGRAMS.

EVENT DRIVEN PROGRAM

VISUAL BASIC USES EVEN DRIVEN PROGRAMMING, EACH ONE TRIGGER ACTIVITIES IN THE PROGRAM EVEN OR ANOTHER. PROGRAM IN CONVENTIONAL PROGRAMMING LANGUAGES RUN FROM THE TOP DOWN. THE CORE VISUAL BASIC PROGRAMMING IS A SET OF INDEPENDENT PIECES OF CODE THAT ARE ACTIVATED BY AND SO RESPOND TO ONLY THE EVENTS THEY HAVE TOLD TO RECOGNIZE. MUCH OF THE PROGRAMMING CODE EVENTS SUCH AS MOUSE CLICKS OCCUR IN WHAT VISUAL BASIC CALLS EVENT PROCEDURE.

WHEN THE APPLICATION IS RUNNING, VISUAL BASIC MONITORS, THE WINDOWS AND THE CONTROLS IN EACH WINDOW FOR ALL THE EVENTS THAT EACH CONTROL CAN RECOGNIZE MOUSE MOVEMENTS, CLICKS, KEYSTROKES, AND SO ON.

WHEN EXAMINE THE APPLICATION TO SEE IF YOU HAVE WRITTEN AN EVENT PROCEDURE FOR THAT EVENT. VISUAL BASIC ALSO PROVIDES SOPHISTICATED ERROR HANDLING FOR ALL TOO COMMON TASK OF PREVENTING USERS FORM BOMBING THE APPLICATION. THIS MEANS THAT ANY CHANGES NEED TO CORRECT THE ROUTINE PROGRAMMING AND TYPOGRAPHICAL ERRORS THAT ARE COMMON WHEN Y0OU BEGIN BUILDING APPLICATION ARE A SNAP.

FORMS

FORM IS THE WINDOW OBJECT ON WHICH THE PROGRAMMER CAN PASTE CONTROLS ON A FORM IS STORED WITH THE FORM IN SEPARATE FILES. GENERAL PROGRAMMING CODE SHARED BY ALL THE FORMS IN THE APPLICATION CAN BE DIVIDED INTO DIFFERENT MODULES THAT ARE ALSO STORED SEPARATELY. TO CRATE USER INTERFACE WE CAN PASTE CONTROLS IN THE FORMS. A PROGRAM CAN HAVE MULTIPLE FORMS IS SAVED AS A SEPARATE FILE IN THE DISK WITH .FRM EXTENSION.

CONTROLS

CONTROLS ARE OBJECTS THAT CAN BE PASTED ON A FORM. CONTROLS ARE AVAILABLE FOR MOST OF THE USER INTERFACE ASPECTS. THEY HAVE PROPERTIES THAT CAN BE SET THROUGH PROPERTIES WINDOW. EACH CONTROL CAN HAVE CODE ATTACHED TO IT THAT WILL BE EVOKED BY THE SPECIFIED EVENT OF THE CONTROL TO WHICH THE CODE IS ATTACHED.

USER CONTROL

THE USER CONTROL OBJECTS IS THE BASE OBJECT USED TO CREATE AN ACTIVE X CONTROL. AN ACTIVE X CONTROL CREATED WITH VISUAL BASIC IS ALWAYS COMPOSED OF A USER CONTROL OBJECT, PLUS ANY CONTROL-REFEREED TO AS CONSTITUENT CONTROLS THAT YOU CHOOSE TO PLACE ON THE USER CONTROL THAT YOU CHOOSE TO PLACE ON THE USER CONTROL LIKE VISUAL BASIC FORMS. USER CONTROLS OBJECTS HAVE CODE MODULES AND VISUAL DESIGNERS. PLACE CONSTITUENT CONTROLS ON THE USER CONTROL OBJECT'S DESIGNERS, JUST AS YOU WOULD PLACE CONTROLS ON A FORM.

THE PROJECT EXPLORER, WHICH DISPLAYS A HIERARCHICAL LIST OF THE CURRENTLY OPEN PROJECTS AND THEIR CONTENTS. THE PROJECT EXPLORER IS A NAVIGATIONAL AND MANAGEMENT TOO ONLY. YOU CANNOT BUILD AN APPLICATION FROM THE PROJECT-EXPLORER. ALL FIRM FILES ASSOCIATED WITH THE PROJECT WIZARDS.

WIZARDS

A TOOL IN THE VISUAL BASIC ENVIRONMENT THAT ASSIST YOU IN THE CREATION OF CUSTOM WIZARDS THAT LOOK AND ACT LIKE THOSE THAT SHIP WITH VISUAL BASIC. THE WIZARD MANAGER MANAGES THE SCREENS OF A WIZARD WHILE IT IS IN THE DEVELOPMENT STATE. THE UTILITY PROVIDES THE INITIAL FRAMEWORK FOR WIZARD TYPE OF ADD-IN AND ABILITY ADD, CHANGE AND DELETE SCREENS FROM THE WIZARD.

VISUAL BASIC 6.0 PROVIDES PROCEDURES THAT CAN LINK WITH THE INTERNET. HTML CAN BE DIRECTLY VIEWED OR PROCESSED FORM VISUAL BASIC ENVIRONMENT. SINCE VISUAL BASIC CAN BE EASILY CONVERTED INTO HTML. VISUAL BASIC IS ALSO COMPATIBLE WITH THIRD PARTY TOOL THAT PROVIDES CONNECTION TO WORLD WIDE WEB.

BOUND DATA CONTROLS

THE TOOLBOX DISPLAYS ALL THE STANDARD VISUAL BASIC CONTROLS, INCLUDING THE DATA CONTROL AND BOUND CONTROLS. IF THE BOUND CONTROLS WE NEED IS NOT DISPLAYED IN THE TOOLBOX, USE THE TOOL CUSTOM CONTROLS DIALOG BOX TO ADD THEM TO OUR PROJECT. TO WRITE CODE THAT REFERENCES THE DATA ACCESS OBJECTS (DAO) THAT THE DATA CONTROL CREATES WE MAY NEED TO USE THE TOOLS REFERENCES DIALOG BOX TO ACTIVATE THE MICROSOFT DAO 2.5 OBJECT LIBRARY. SOME OF THE CONTROL LIST BELOW ALSO HAS LIBRARIES THAT REQUIRE ACTIVATION USING WITH VISUAL BASIC. USE TO PROVIDE ACCESS TO DATA IN DATABASE THROUGH BOUND CONTROLS ON CREATES AND MANAGE DATABASE AND RECORD SET OBJECTS FOR USE BY BOUND CONTROLS.

DBCOMBO

USE TO DRAW A BOUND COMBINATION LIST BOX AND TEXT BOX. THE LIST CAN BE FILLED AUTOMATICALLY FROM A DATA CONTROL. THE USER CAN EITHER CHOOSE AN ITEM FROM THE LIST OR ENTER A VALUE IN THE TEXT BOX. CAN BE USED TO PROVIDE READ/WRITE ACCESS TO A SPECIFIED TEXT DATA FIELD SELECTED FROM THE LIST.

DBLIST

USE TO DISPLAY A DATA CONTROL GENERATED LIST OF ITEMS FROM WHICH THE USER CAN CHOOSE ONE. THE LIST CAB BE FILLED AUTOMATICALLY FROM A DATA CONTROL AND PROVIDE READ/WRITE ACCESS TO A SPECIFIED TEXT DATA FIELD SELECTED FROM THE LIST.

DBGRID

USE TO DRAW A BOUND COMPOSED OF MULTIPLE RECORDS. THE GRID CAN BE FILLED AUTOMATICALLY FROM A DATA CONTROL. THE USER CAN EITHER CHOOSE AN ITEM FROM THE GRID, OR ENTER A VALUE IN THE NEW RECODES. CAN BE USED TO PROVIDE READ/WRITE ACCESS TO A RECORD SET.

LABEL

USE FOR TEXT THAT WE DON'T WANT THE USER TO CHANGE, SUCH AS A CAPTION UNDER A GRAPHICS. CAN BE USED TO PROVIDE READ/WRITE ACCESS TO A SPECIFIED TEXT DATA FIELD.

TEXT BOX

USE TO HOLD TEXT THAT THE USER CAN EITHER OR CHANGE OR CHANGE. CAN BE USED TO PROVIDE READ/WRITE ACCESS TO A SPECIFIED TEXT DATA FIELD.

CHECK BOX

USER TO CREATE A BOX THAT THE USER CAN EASILY CHOOSE TO INDICATE IF SOMETHING IS TRUE OR FALSE, OR TO DISPLAY MULTIPLE CHOICE WHEN THE USER CAN CHOOSE MORE THAN ONE. CAN BE USED TO PROVIDE READ/WRITE ACCESS TO A SPECIFIED BOOLEAN OR BIT DATA FIELD.

COMBO BOX

USE TO DRAW A COMBINATION LIST BOX AND TEXT BOX. THE LIST IS FILLED WITH ADD ITEM METHOD. THE USER CAN EITHER CHOOSE AN ITEM FROM THE LIST OR ENTIRE VALUE IN THE TEXT BOX. CAN BE USED TO PROVIDE READ/WRITE ACCESS TO SPECIFY TEXT DATA SELECTED FROM THE LIST.

LIST BOX

USE TO DISPLAY A LIST OF ITEMS FROM WHICH THE USER CAN CHOOSE ONE. THE LIST IS FILLED WITH THE ADD ITEM METHOD. THE USER CAN CHOOSE AN ITEM FROM THE LIST. CAN BE USED TO PROVIDE READ/WRITE ACCESS TO A SPECIFIED TEXT DATA FIELD SELECTED FROM THE LIST.

PICTURE BOX

USE TO DISPLAY A GRAPHICAL IMAGE FROM A BITMAP, ICON, OR METAFILE ON OUR FORM. CAN BE USED TO PROVIDE READ/WRITE ACCESS TO A SPECIFIED IMAGE. BINARY DATA FILED.

IMAGE

USE TO DISPLAY A GRAPHICAL IMAGE FROM A BITMAP, ICON, OR METAFILE ON OUR FORM, IMAGES DISPLAYED IN AN IMAGE CONTROL ARE ONLY DECORATIVE AND USE FEWER RESOURCES THAT A PICTURES BOX. CAN BE USED TO PROVIDE READ/WRITE ACCESS TO A SPECIFIED IMAGE/BINARY DATA FILED.

ACTIVEX CONTROL

AN ACTIVEX CONTROL IS AN EXTENSION TO THE VISUAL BASIC TOOLBOX. WE CAN USE ACTIVEX CONTROLS JUST LIKE THE STANDARD CONTROLS IN TOOLBOX SUCH AS CHECKBOX. WHEN WE ADD AN ACTIVEX CONTROL TO A PROGRAM, IT BECOMES PART OF THE DEVELOPMENT AND RUN-TIME ENVIRONMENT AND PROVIDES NEW FUNCTIONALITY FOR YOUR APPLICATION.

ACTIVEX CONTROLS LEVERAGE OUR CAPABILITIES AS A VISUAL BASIC PROGRAMMER BY RETAINING SOME FAMILIAR PROPERTIES, EVENTS, AND METHODS, SUCH AS THE NAME PROPERTY, WHICH BEHAVES AS WE EXPECT.

USING THE WINSOCK (ACTIVE X) CONTROLS

THE WINSOCK CONTROL IS AN ACTIVEX CONTROL THAT PROVIDES A WAY FOR APPLICATIONS TO COMMUNICATE USING THE TCP/IP OR IRDA PROTOCOLS. A WINSOCK CONTROL USES THE UNDERLYING NETWORK CONNECTION OR INFRARED PORT TO TRANSFER DATA.

BECAUSE A WINSOCK CONTROL CAN ACT AS A CLIENT THAT CONNECTS TO A SERVER APPLICATION OR AS A SERVER THAT PROVIDES CONNECTIONS TO NETWORK CLIENTS, THE FIRST STEP IN USING A WINSOCK CONTROL IS TO DETERMINE WHETHER THE CONTROL WILL ACT AS A CLIENT OR AS A SERVER.

USING THE COMMON DIALOG (ACTIVE X) CONTROLS

THE COMMON DIALOG CONTROL PROVIDES A STANDARD SET OF DIALOG BOXES FOR OPERATIONS SUCH AS OPENING AND SAVING FILES, SETTING PRINT OPTIONS, AND SELECTING COLORS AND FONTS. THE CONTROL ALSO HAS THE ABILITY TO DISPLAY HELP BY RUNNING THE WINDOWS HELP ENGINE.

THE COMMON DIALOG CONTROL PROVIDES AN INTERFACE BETWEEN VISUAL BASIC AND THE PROCEDURES IN THE MICROSOFT WINDOWS DYNAMIC-LINK

LIBRARY COMMDLG.DLL. TO CREATE A DIALOG BOX USING THIS CONTROL, COMMDLG.DLL MUST BE IN YOUR MICROSOFT WINDOWS \SYSTEM DIRECTORY

USING THE RICHTEXTBOX (ACTIVE X) CONTROLS

THE RICHTEXTBOX CONTROL ALLOWS THE USER TO ENTER AND EDIT TEXT WHILE ALSO PROVIDING MORE ADVANCED FORMATTING FEATURES THAN THE CONVENTIONAL TEXTBOX CONTROL.

THE RICHTEXTBOX CONTROL PROVIDES A NUMBER OF PROPERTIES YOU CAN USE TO APPLY FORMATTING TO ANY PORTION OF TEXT WITHIN THE CONTROL. USING THESE PROPERTIES, YOU CAN MAKE TEXT BOLD OR ITALIC, CHANGE ITS COLOR, AND CREATE SUPERSCRIPTS AND SUBSCRIPTS. YOU CAN ALSO ADJUST PARAGRAPH FORMATTING BY SETTING BOTH LEFT AND RIGHT INDENTS, AS WELL AS HANGING INDENTS.

USING THE MS COMMON (ACTIVEX) CONTROLS

THE STATUS BAR IS ONE OF THE ACTIVEX CONTROLS IN MICROSOFT COMMON CONTROLS. A STATUS BAR CONTROL PROVIDES A WINDOW, USUALLY AT THE BOTTOM O F A PARENT FORM, THROUGH WHICH AN APPLICATION CAN DISPLAY VARIOUS KINDS OF STATUS DATA. THE STATUS BAR CAN BE DIVIDED UP INTO A MAXIMUM OF SIXTEEN PANEL OBJECTS THAT ARE CONTAINED IN A PANEL'S COLLECTION. OUR FORM CONTAINS DB OPERATION (ADD, MODIFY AND DELETE) PANELS.

DATA MANAGEMENT

IN VISUAL BASIC DATA CAN BE STORED AS FILES OR AS DATABASES. DATA STORAGE USING DATABASE TECHNOLOGY IS THROUGH THE VISUAL BASIC CONTROL CALLED DATA CONTROL. IT CAN INCORPORATE DATABASES GENERATED USING THE DATABASE MANAGEMENT SYSTEM CONNECTED TO THE PROGRAMMING

ENVIRONMENT OR EXTERNAL DATABASES. VISUAL BASIC LETS YOU CREATE AND MANIPULATE DATA ACCESS OBJECTS SUCH AS DATABASE, FIELD AND INDEX OBJECTS. WE CAN USE THE PROPERTIES AND METHODS OF THESE OBJECTS TO PERFORM OPERATIONS OF THE DATABASE.

DAO LETS US VIEW PROGRAMS THAT ACCESS MANY DIFFERENT KINDS OF DATABASES INCLUDING NATIVE DATABASES THAT USE THE SAME FORMAT AS MSACCESS, EXTERNAL DATABASE LIKE FOXPRO, DBASE ETC. AND ODBC CLIENT/SERVER DATABASES LIKE ORACLE.

DATA ACCESS OVERVIEW

VISUAL BASIC ENABLES US TO ACCESS AND MANIPULATE DATA IN DATABASES AND TO MANAGE DATABASE, THEIR OBJECTS AND STRUCTURE. WHEN CREATING PROGRAMS TO ACCESS DATABASES, WE HAVE A VARIETY OF DEVELOPMENT OPTIONS, WE CAN USE.

THE BUILD IN MICROSOFT JET DATABASE ENGINE WITH VISUAL BASIC DATA CONTROL. THE JET ENGINE WITH VISUAL BASIC DATA ACCESS OBJECT (DAO). THE ODBC API TO ACCESS ANY EXTERNAL DATABASE WITH AN ACCESS MICROSOFT SQL SERVER.

WE CAN USE VISUAL BASIC DATA ACCESS OBJECTS OR THE DATA CONTROL TO MANIPULATE DATABASES IN THE NATIVE JET ENGINE. MDB FORMAT. IN ADDITION WE CAN USE THE JET DATABASES THAT CAN BE ACCESSED WITH ODBC

DRIVER USING THE SAME DAP CODE OF THE DATA CONTROL. THE JET ENGINE VERSION 2.5 IS SUPPLIED WITH VISUAL BASIC SO YOU DON'T NEED ANY ADDITIONAL DLLS OF DRIVER.

DATA ACCESS OBJECTS

BOTH THE STANDARD THE PROFESSIONAL EDITION OF VISUAL BASIC SUPPORT DATA ACCESS OBJECTS (DAO) SUCH AS THE DBENGINE, WORKSPACE,

DATABASE, TABLEDEF, FIELD, INDEX AND RECORDSET OBJECTS. ALTHOUGH WE CAN'T DECLARE NEW DATA ACCESS OBJECTS WITH STANDARD A EDITION, WE CAN MANIPULATE THOSE CREATED FOR US BY DATA ACCESS OBJECTS, SUCH AS TABLED OF OBJECTS THAT DEFINE TABLES, FIELDS, OBJECTS THAT DEFINE FIELDS, AND SO ON. EACH OBJECT HAS PROPERTIES THAT ARE COLLECTIONS OF OTHER OBJECTS. IN THE SAME WAY THAT A FORM HAS A CONTROLS COLLECTION, A DATABASE HAS TABLEDEF COLLECTION OF TABLEDEF OBJECTS, AND A TABLEDEF HAS COLLECTION OF FIELD OBJECT. WHEN YOU WRITE CODE SETTING ESTABLISHED AND MANAGED BY THE JET ENGINE.

EXECUTION OF THE PROJECT

A NEW PROJECT CAN BE SAVED BE SELECTING SAVE PROJECT OF BY CLICKING THE SAQVE PROJECT TOOL. BUTTON N THE TOOL BAR TO DISPLAY THE SAVE FILES AS DIALOG BOX. THE PROGRAM CAN BE RUN BY PRESSING F5 OR BY CLICKING THE TOOL BUTTON. VISUAL BASIC INCLUDES A BUILT IS COMMAND THAT TERMINATES THE PROGRAM THE END STATEMENT.

QUICK SORTING

IN TABLE AND FORM DATASHEETS AND IN FORMS, YOU CAN CLICK THE SORT ASCENDING OR SORT DESCENDING BUTTONS ON THE TOOLBAR TO QUICKLY SORT THE CURRENT RECORDS BASED ON THE SELECTED COLUMN.

NEW PROPERTY BUILDERS AND THE BUILD BUTTON

MANY PROPERTIES NOW INCLUDE BUILDERS THAT HELP YOU SET THE PROPERTY. FOR EXAMPLE, YOU CAN USE THE EXPRESSION BUILDER FOR PROPERTIES THAT CAN EXPRESSIONS, THE QUERY BUILDER FOR THE RECORD SOURCE AND ROW SOURCE PROPERTIES, AND THE COLOR BUILDER FOR DEFINING BACK COLOR, BORDER COLOR, AND FORE COLOR PROPERTIES. SOME PROPERTIES

ALLOW YOU TO SELECT THE EXPRESSION BUILDER, THE MACRO BUILDER, OR THE CODE BUILDER.

WHEN YOU SELECT A PROPERTY WITH A BUILDER, MICROSOFT ACCESS DISPLAYS A BUILD BUTTON NEXT TO THE PROPERTY BOX. CLICK THE BUTTON TO START THE BUILDER. A BUILD BUTTON ALSO APPEARS ON SOME TOOLBARS.

MS ACCESS AND ITS FEATURES

WHAT NEW IN MS ACCESS?

MICROSOFT ACCESS VERSION 7.0 INCLUDES MANY NEW FEATURES AND ENHANCEMENTS THAT HELP YOU WORK WITH YOUR DATABASES. A LOW OF THESE NEW FEATURES ARE SUMMARIZED BELOW MOVABLE CUSTOMIZABLE TOOLBARS.

MICROSOFT ACCESS TOOLBARS ARE NOW MOVEABLE AND CUSTOMIZABLE. YOU CAN MOVE TOOLBARS AROUND IN THE MICROSOFT ACCESS WINDOWS HIDE THEM INDIVIDUALLY, AND CUSTOMIZE THEM BY ADDING BOTTOMS. YOU CAN ALSO CREATE YOUR OWN TOOLBARS AND ATTACH THEM TO FORMS AND REPORTS.

TOOL TIPS

MOVE THE POINTER ONTO THE FACE OF ANY BUTTON, INCLUDING THOSE IN THE TOOLBOX OR PALETTE, TO DISPLAY THE NAME OF THE BUTTON OR OPTION.

SHORTCUT MENUS

PRESS THE RIGHT MOUSE BUTTON WHEN YOU ARE DEFINING A CONTROL, SETTING A PROPERTY, OR PERFORMING MOST ANY OTHER TASK IN MICROSOFT ACCESS TO DISPLAY A SHORTCUT MENU. COMMANDS APPROPRIATE TO THE CURRENT TASK APPEAR ON THE MENU.

INPUT MASKS

YOU CAN MAKE DATA ENTRY QUICKER AND MORE ACCURATE BY ADDING INPUT MASKS TO FIELDS THAT REQUIRE THAT DATA BE ENTERED IN THE SAME WAY

EVERY TIME. IN A PHONE NUMBER FIELD, FOR EXAMPLE, YOU COULD ADD AN INPUT MASK THE AUTOMATICALLY ADDS THE NECESSARY PARENTHESES, HYPHENS, AND SPACES.

IMPROVED SELECTION IN DATASHEETS

YOU CAN SELECT AND COPY TO THE CLIPBOARD A BLOCK OF ADJACENT CELLS WITHIN A DATASHEET. YOU SELECT THE DATA YOU WANT BY DRAGGING THE POINTER THOUGH THE CELLS.

DESIGN METHODOLOGY

DESIGN IS CONCERNED WITH IDENTIFYING SOFTWARE COMPONENTS, SPECIFYING RELATIONSHIPS AMONG COMPONENTS, SPECIFYING SOFTWARE STRUCTURE AS PROVIDING A BLUE PRINT FOR THE IMPLEMENTATION PHASE. THE DESIGN CONSISTS OF THREE TYPES.

- ARCHITECTURAL DESIGN
- DETAILED DESIGN
- EXTERNAL DESIGN

ARCHITECTURAL DESIGN

IT INVOLVES IDENTIFYING THE SOFTWARE COMPONENTS, DECOUPLING AND DECOMPOSING THEM INTO PROCESSING MODULES AND CONCEPTUAL DATA STRUCTURE AND SPECIFYING RELATIONSHIPS AMONG THE COMPONENTS.

DETAILED DESIGN

THE DESIGN SPEAKS ABOUT THE DETAILS OF HOW THE PACKAGE THE PROCESSING MODULES AND HOW TO IMPLEMENT THE PROCESSING ALGORITHMS, DATA STRUCTURES AND INTER CONNECTING AMONG MODULES AND DATA STRUCTURE.

EXTERNAL DESIGN

EXTERNAL DESIGN OF SOFTWARE INVOLVES CONCEIVING, PLANNING OUT, AND SPECIFYING THE EXTERNALLY OBSERVABLE CHARACTERISTICS OF A SOFTWARE PRODUCT.

INPUT DESIGN

INPUTS THAT ARE REQUIRED FOR THE VARIOUS PROCESSES THAT ARE TO BE CAREFULLY ANALYZED AND CARE HAVE TO TAKEN TO AVOID RECURRING OF THE SAME INPUTS. INPUT DESIGN IS THE PROCESS OF CONVERTING USER ORIGINATES INPUTS INTO COMPUTER-BASED FORMAT. THE GOAL OF DESIGNING INPUT DATA IS TO MAKE DATA ENTRY AS EASY AS POSSIBLE AND ERROR FREE. WELL-DESIGNED INPUTS SERVE FOUR PURPOSES TO CONTROL WORKFLOW, TO REDUCE REDUNDANCIES IN RECORDING DATA, TO ALLOW EASIER CHECKING OF DATA AND TO INCREASE CLERICAL ACCURACY. WHEN DATA IS KEYED INTO THE SYSTEM, THE OPERATOR MUST RECEIVE THE DATA IN A FORM THAT CAN BE EASILY UNDERSTOOD. IT SHOULD BE SELF-EXPLANATORY AND PROVIDE THE SUFFICIENT INFORMATION TO THE USER FOR EASE OF ENTRY OF INPUTS.

FORMS ARE DESIGNED FOR RETRIEVING INPUTS FROM THE USER. IT IS USED TO ENTER DATA AND IT ALLOWS CORRECTING THE INCORRECT ENTRY OF DATA. THE SYSTEM IS A MENU DRIVEN ONE. THIS SIMPLIFIES THE COMPUTER DATA ACCESS OR DATA ENTRY. THE DATA THAT CAN BE ACCESSED BY EACH USER CAN BE SPECIFIED SO THAT, THE REPORTS WILL BE RESTRICTED TO THAT LEVEL ONLY. IN FACT THE SYSTEM ALLOWS THE DEFINITION OF DATA ACCESS RIGHTS FOR EACH USER FOR EACH FUNCTION. THIS ENSURES THAT ONLY THE RIGHT USER GETS THE INFORMATION.

INPUT DESIGN FOR FORM SEND MAIL

IN THIS FORM WE GET THE E-MAIL ID OF THE SENDER AND RECEIVER, CONNECT IT TO THE INTERNET WITH THE HELP OF SMTP (SIMPLE MAIL TRANSFER PROTOCOL) HOST.

HERE WE ALSO GET DETAILS FROM THE USER SUCH AS THE RECEIVERS MAIL ADDRESS, SUBJECT, MESSAGE AND ATTACHMENTS IF ANY.

INPUT DESIGN FOR FORM RECEIVE MAIL

IN THIS FORM WE GIVE THE COMPANY'S MAIL ID AND CONNECT TO THE INTERNET USING POP3 (POST OFFICE PROTOCOL).

THEN WE RECEIVE THE E-MAIL AND FORWARD IT TO THE RESPECTIVE CLIENTS.

INPUT DESIGN FOR BLOCKING APPLICATIONS

HERE WE ADD THE FILES TO BE BLOCKED USING COMMON DIALOG CONTROLS SHOWOPEN PROPERTY.

THE APPLICATIONS ARE BLOCKED USING AN USER CONTROL IN THE DISABLE.OCX FILE.

KEYSTROKE DYNAMICS

WE USE KEY STROKE DYNAMICS FOR SECURITY PURPOSE. WE AUTHENTICATE USERS WITH THE HELP OF KEYSTROKE DYNAMICS. KEYSTROKE DYNAMICS IS AUTHENTICATING THE USERS WITH THE HELP OF THEIR TYPING SPEED, ALONG WITH THEIR TYPING SPEED.

CODE DESIGN

A CODE IS AN ORDERED COLLECTION OF SYMBOLS DESIGNED TO PROVIDE UNIQUE IDENTIFICATION OF ANY ENTITY OR ATTRIBUTE. THE PURPOSE OF CODE IS TO FACILITATE THE IDENTIFICATION AND RETRIEVAL OF INFORMATION.

TYPE OF CODE

A COMMON PURPOSE OF CODE CLASSIFIES OBJECTS FOR ANALYSIS. THEY ARE ALSO USED FOR ECONOMY OF CONTENT AND FORMAT FOR IDENTIFYING, ACCESSING, SORTING AND MATCHING RECORDS.

TWO MAIN TYPES OF CODE ARE

- SIGNIFICANT CODE
- NON SIGNIFICANT CODE

SIGNIFICANT CODE

A SIGNIFICANT CODE IS ONE THAT AS WELL AS PROVIDE UNIQUE IDENTIFICATION, IS DESIGNED TO FURNISH ADDITIONAL MEANING, WHICH CAN YIELD LOGICAL COLLATING OR PNEUMONIC CODING SIGNIFICANCE.

NON-SIGNIFICANT CODE

A NON-SIGNIFICANT CODE IS ONE IN WHICH INDIVIDUAL VALUES ARE MEANINGLESS WITHOUT SOME DEFINED RELATIONSHIP TO THE ENTRY OR ATTRIBUTES, WHICH ARE BEING CODED. SO, THEY ARE ASSIGNED TO PROVIDE UNIQUE IDENTIFICATION.

DATABASE DESIGN

CHECKMAIL

Field Name	Data Type
Number	Number
Date	Date/Time
From_Address	Text
Subject	Text
Message	Memo
Attach	Text
Status	Yes/No

APPOINT

Field Name	Data Type
Date	Text
Emp_ID	Text
Emp_Name	Text
Emp_Dsg	Text
Emp_OutTime	Text
Emp_InTime	Text
Emp_Status	Text

BLOCK

Field Name	Data Type
filename	Memo

EMP_DET

Field Name	Data Type
Emp_ID	Text
Emp_Name	Text
Designation	Text
DOB	Text
DOJ	Text
Age	Number
Sex	Text
Address	Memo
Phone_Resi	Text
Phone_Off	Text
E_mail	Text

FORWARD

Field Name	Data Type
no	Print Preview
Date	Date/Time
From_Add	Text
Subject	Text
Message	Memo
Attach	Text

SEND

Field Name	Data Type
No	Number
Date	Date/Time
From_Add	Text
To_Add	Text
Subject	Text
Message	Memo
Attach	Text

LOG

Field Name	Data Type
S_no	Number

MSG

Field Name	Data Type
S_No	Number
Message	Memo

PASS

Field Name	Data Type
EmpID	Text
Password	Text
Speed	Number

THUS THE TABLE DESIGNS ARE PRESENTED ABOVE.

OUTPUT DESIGN

THE MAIN IDEA OF DEVELOPING THIS SOFTWARE SYSTEM IS TO GENERATE VARIOUS OUTPUTS IN NECESSARY FORMAT, WHICH WILL AID IN PLANNING AND DECISION-MAKING. THE OUTPUTS SHOULD INCLUDE ALL THE NECESSARY DETAILS AND THE REQUIRED INFORMATION. THE PRIMARY CONSIDERATION IN OUTPUT DESIGN IS TO ARRANGE THE DATA IN A FORM, WHICH IS CONVENIENT TO THE USER. THE LAYOUTS OF THE FORM SHOULD BE PLEASING. CARE SHOULD BE TAKEN THAT THE PROMPT AND THE ICONS ARE POSITIONED AT THE CORRECT PLACE. ALSO THE SIZE OF THE FORM SHOULD BE APPROPRIATE DEPENDING ON ITS CONTENTS. WHENEVER ERROR MESSAGES ARE DISPLAYED IT SHOULD BE AS LONG AS POSSIBLE AND MEANINGFUL. ALL HEADERS AND DISPLAYS SHOULD BE RELEVANT TO THE MESSAGE. ERROR MESSAGE SHOULD NOT CONTAIN ANY PROGRAMMING RELATED TERMINOLOGY.

SYSTEM TESTING

THIS IS THE CRITICAL ELEMENT OF SOFTWARE QUALITY ASSURANCE. IT REPRESENTS THE ULTIMATE REVIEW OF SPECIFICATION, DESIGN AND CODING OF SOFTWARE. TESTING IS CALLED A DESTRUCTIVE ACTIVITY. IT IS A PROCESS OF EXECUTING A PROGRAM WITH THE INTENT OF FINDING ERRORS. GOOD TESTING IS THAT WHICH HAS THE HIGH PROBABILITY OF FINDING AN ERROR WHICH IS YET UNDISCOVERED. A SUCCESSFUL TEST UNCOVERS A YET UNDISCOVERED ERROR IN THE SOFTWARE. THE FINAL GOAL OF TESTING IS TO SEE THAT THE SYSTEM PERFORMS ITS INTENDED PURPOSE SATISFACTORILY. THIS SYSTEM HAS UNDERGONE VARIOUS STAGES FOR VALIDATION OF RESULTS AND FOR ITS INTEGRITY.

UNIT TESTING

THIS TEST IS USED TO TRACE OUT THE BUGS IN MINIMAL PART OF THE CODE. IT IS DONE DURING THE CODE DEVELOPMENT. THE GOAL IS TO TEST THE INTERNAL LOGIC OF THE MODULE.

INTERFACE TESTING

THIS IS USED TO ASSURE THAT THE INFORMATION PROPERLY FLOWS INTO AND OUT OF THE PROGRAM UNITS UNDER THE TEST. ALL INDEPENDENT PATHS THROUGH THE MODULE OPERATE PROPERLY AT BOUNDARIES ESTABLISHED TO LIMIT OR RESTRICT PROCESSING.

CODE TESTING

IN ORDER TO TEST THE SYSTEM BOTH THE CODE TESTING AND THE PROGRAM IS EXECUTED WITH DEVELOPED TEST CASES. IN THE ENQUIRIES, ESTIMATION AND QUOTES AND CUSTOMER ORDER MODULE EACH AND EVERY SUB MODULE WAS TESTED WITH DEVELOPED TEST CASES IN ORDER TO GET THE OUTPUT WITHOUT ANY BREAK IN THE PATH OF EXECUTION.

INTEGRATION TESTING

IN THIS PROJECT, INTEGRATION TESTING IS VERY IMPORTANT. INTEGRATION TESTING IS A SYSTEMATIC TECHNIQUE FOR CONSTRUCTING A PROGRAM STRUCTURE. THOUGH EACH PROGRAM WORKS INDEPENDENTLY, THEY SHOULD ALSO WORK AFTER LINKING THEM TOGETHER. IN INTEGRATION TESTING MANY TESTED MODULES ARE COMBINED INTO SUBSYSTEMS, WHICH ARE THEN TESTED. THE GOAL IS TO SEE IF THE MODULES CAN BE INTEGRATED PROPERLY, THE EMPHASIS BEING ON TESTING INTERFACE BETWEEN MODULES.

VALIDATION TESTING

TO UNCOVER FUNCTIONAL ERRORS, THAT IS TO CHECK WHETHER THE FUNCTIONAL CHARACTERISTICS CONFORM TO THE SPECIFICATION OR NOT.

SYSTEM SECURITY

A COMPUTER - BASED SYSTEM IS A COMBINATION OF MANY ASSETS OR RESOURCES DESIGNED TO PERFORM SOME FUNCTION OR TO PROVIDE SOME SERVICES. A COMPUTER SYSTEM IS SECURED AGAINST A PARTICULAR THREAD IF COUNTER MEASURES HAVE BEEN TAKEN TO REDUCE TO AN ACCEPTABLY LOWER AMOUNT THAT THE THREAT MAY BE EXPECTED TO CAUSED OVER A GIVEN PERIOD OF TIME. SECURITY IS CRITICAL IN SYSTEM DEVELOPMENT.

THE SYSTEM SECURITY PROBLEM CAN BE DIVIDED INTO FOUR RELATED ISSUES.

1. SYSTEM SECURITY REFERS TO THE TECHNICAL INNOVATION AND PROCEDURES APPLIED TO THE HARDWARE AND THE OPERATING SYSTEM TO PROTECT AGAINST ACCIDENTAL DAMAGE FROM A DEFINED THREAT.

2. SYSTEM INTEGRITY REFERS TO THE PROPER FUNCTION OF HARDWARE AND PROGRAMS, APPROPRIATE PHYSICAL SECURITY AND SAFETY AGAINST EXTERNAL THREAT SUCH AS EAVESDROPPING AND WIRETAPPING.

3. PRIVACY DEFINES THE RIGHTS OF THE USER OR ORGANIZATION TO DETERMINE WHAT INFORMATION THEY ARE WILLING TO SHARE WITH OR ACCEPT FROM OTHERS AND HOW THE ORGANIZATION CAN BE PROTECTED AGAINST UNFAIR OR EXCESSIVE DISCRIMINATION OF INFORMATION ABOUT IT.

4. CONFIDENTIALITY IS A SPECIAL ATTENTION TO BE GIVEN TO THE SENSITIVE INFORMATION IN A DATABASE TO MINIMIZE THE POSSIBLE INVASION OF

INTRUDERS. IT IS AN ATTRIBUTE OF INFORMATION THAT CHARACTERIZES ITS NEED OF PROTECTION.

ACCEPTANCE TESTING

THE LAST LEVEL OF TESTING CONDUCTED IS THE ACCEPTANCE TESTING. ACCEPTANCE TESTING IS SOMETIMES PERFORMED WITH REALISTIC DATA OF THE CLIENT TO DEMONSTRATE THAT THE SOFTWARE IS WORKING SATISFACTORILY. TESTING HERE FOCUSES ON THE EXTERNAL BEHAVIOR OF THE SYSTEM; THE INTERNAL LOGIC OF THE PROGRAM IS NOT EMPHASIZED.

SYSTEM IMPLEMENTATION

SYSTEM IMPLEMENTATION IS THE PROCESS OF MAKING THE NEWLY DESIGNED SYSTEM FULLY OPERATIONAL AND CONSISTENT IN PERFORMANCE. THAT IS, IMPLEMENTATION IS THE PROCESS OF HAVING THE PERSONNEL CHECK OUT AND PUT NEW EQUIPMENT INTO USE, TRAIN THE USER IS TO USE THE NEW SYSTEM AND CONSTRUCT ANY FILE THAT IS NEEDED TO USE IT. AT THIS STAGE THE MAIN WORKLOAD, THE GREATEST UPHEAVAL AND THE MAJOR IMPACT ON THE EXISTING PRACTICES SHIFT TO THE USER DEPARTMENT. IF THE IMPLEMENTATION IS NOT CAREFULLY PLANNED AND CONTROLLED, IT CAN CAUSE CHAOS. THUS IT CAN BE CONSIDERED TO BE THE MOST CRUCIAL STAGE IN ACHIEVING A SUCCESSFUL NEW SYSTEM AND IN GIVING THE USERS CONFIDENCE THAT THE NEW SYSTEM WILL WORK AND BE EFFECTIVE.

BEFORE THE DEVELOPMENT OF THE SYSTEM, THE USER SPECIFICATIONS, THE FORMS AND THE VALIDATIONS BASED ON THE FORMS AND THE RESPECTIVE

REPORTS ARE PREPARED. THE USER CAN SPECIFY THE CHANGES IF ANY, THEN THE DESIGN DEPARTMENT EXAMINE THE CHANGES AND IF ACCEPTED THEN THE REQUIREMENT OF THE USER ARE TAKEN CARE OF. THIS IS THE STAGE WHERE THE SYSTEM DESIGN BEGINS. THAT IS THE THEORETICAL DESIGN IS CONVERTED INTO A WORKING SYSTEM. A MOCK DATA SHEET IS PREPARED WHICH CONTAINS THE RESULTS FOR EACH FORM. ALL THE TECHNICAL ERRORS ARE FIXED AND THE TEST DATA IS ENTERED. THEN THE REPORTS ARE PREPARED AND COMPARED WITH THAT OF THE EXISTING SYSTEM. IF THE NEW SYSTEM IS NOT WORKING PROPERLY, THEN ONCE AGAIN WE CAN GO BACK TO THE EXISTING SYSTEM AND AFTER RECTIFICATION; THE NEW SYSTEM CAN BE INSTALLED.

GOOD DOCUMENTATION, ALTHOUGH ESSENTIAL, DOESN'T REPLACE TRAINING. THERE IS NO SUBSTITUTE FOR HANDS ON OPERATION OF THE SYSTEM. VENDORS, IN SERVICE TRAINING, ON - SITE AND IN - HOUSE TRAINING ARE THE VARIOUS TYPES OF TRAINING. THE USERS ARE OBSERVED OVER A PERIOD OF TIME AND ALL THE PROBLEMS ENCOUNTERED DURING THIS STAGE ARE TAKEN CARE OF AND THE SYSTEM IS AGAIN UPDATED IN ORDER TO MEET THE CUSTOMER'S REQUIREMENTS.

CONCLUSION

CONCLUSION

THE PROJECT WORK ENTITLED '**MAIL SERVER MANAGEMENT AND SECURITY USING KEYSTROKE DYNAMICS**' DESCRIBED SO FAR HAS BEEN DESIGNED, TESTED AND DOCUMENTED COMPLETELY. THE SYSTEM HAS BEEN DEVELOPED TO OVERCOME THE PROBLEMS WITH THE EXISTING SYSTEMS.

SCOPE FOR FURTHER WORK

SCOPE FOR FURTHER WORK

- IN FUTURE WE WILL CHECK THE AVAILABILITY OF THE INDIVIDUAL BY CALLING TASK SCHUDELER.
- WE CAN ADD MORE FEATURES RELETAD TO OUR PROJECT.

BIBLIOGRAPHY

BIBLIOGRAPHY

TEXT BOOKS:

GARY CORNELL	VISUAL BASIC 6.0
DAVID JUNG	VISUAL BASIC 6.0 SUPER BIBLE
EVANGELOS PETROUTSOS	MASTERING VISUAL BASIC 6
LEE	INTRODUCTION TO SYSTEM ANALYSIS AND DESIGN
STEVE BROWN	PROFESSIONAL FEATURES OF VISUAL BASIC 6.0
RICHARD E. FAIRLEY	SOFTWARE ENGINEERING CONCEPTS
PRESSMAN	SOFTWARE ENGINEERING

REFERENCE MANUALS:

MICROSOFT WINDOWS 2000 PROGRAMMERS GUIDE

MICROSOFT VISUAL BASIC 6.0 PROGRAMMERS GUIDE

APPENDIX

APPENDIX – A: Sample Coding

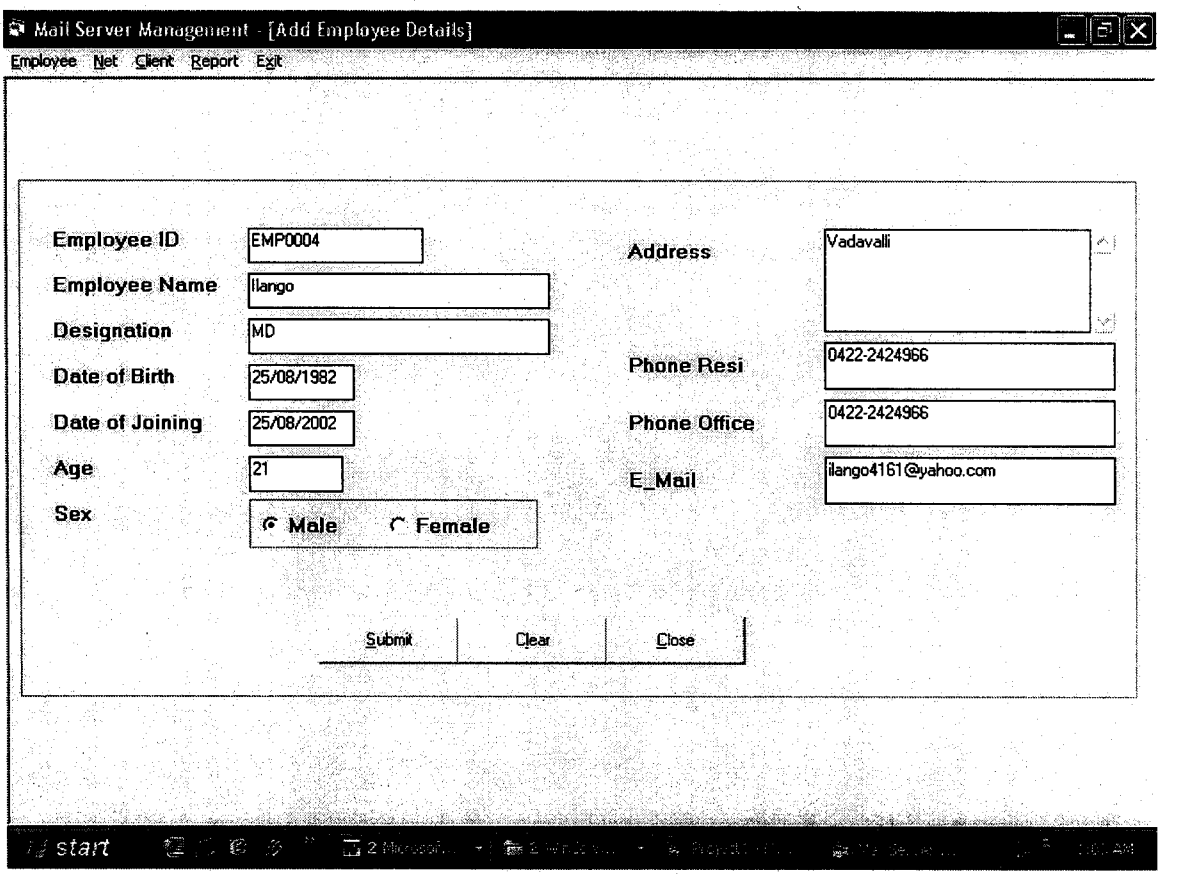
```
Public Function UUDecodeToFile(strUUCodeData As String, strFilePath As
String)

  Dim vDataLine   As Variant
  Dim vDataLines  As Variant
  Dim strDataLine As String
  Dim intSymbols  As Integer
  Dim intFile     As Integer
  Dim strTemp     As String
  If Left$(strUUCodeData, 6) = "begin " Then
  strUUCodeData = Mid$(strUUCodeData, InStr(1, strUUCodeData, vbLf) + 1)
  End If
  If Right$(strUUCodeData, 5) = "end" + vbCrLf Then
     strUUCodeData = Left$(strUUCodeData, Len(strUUCodeData) - 10)
  End If
  intFile = FreeFile
  Open strFilePath For Binary As intFile
     vDataLines = Split(strUUCodeData, vbCrLf)
     For Each vDataLine In vDataLines
           strDataLine = CStr(vDataLine)
           intSymbols = Asc(Left$(strDataLine, 1)) - 32
           strDataLine = Mid$(strDataLine, 2)
           strDataLine = Replace(strDataLine, "`", " ")
           For i = 1 To Len(strDataLine) Step 4
              strTemp = strTemp + Chr((Asc(Mid(strDataLine, i, 1)) - 32) * 4 + _
                    (Asc(Mid(strDataLine, i + 1, 1)) - 32) \ 16)
     strTemp = strTemp + Chr((Asc(Mid(strDataLine, i + 1, 1)) Mod 16) * 16 + _
                    (Asc(Mid(strDataLine, i + 2, 1)) - 32) \ 4)
     strTemp = strTemp + Chr((Asc(Mid(strDataLine, i + 2, 1)) Mod 4) * 64 + _
```

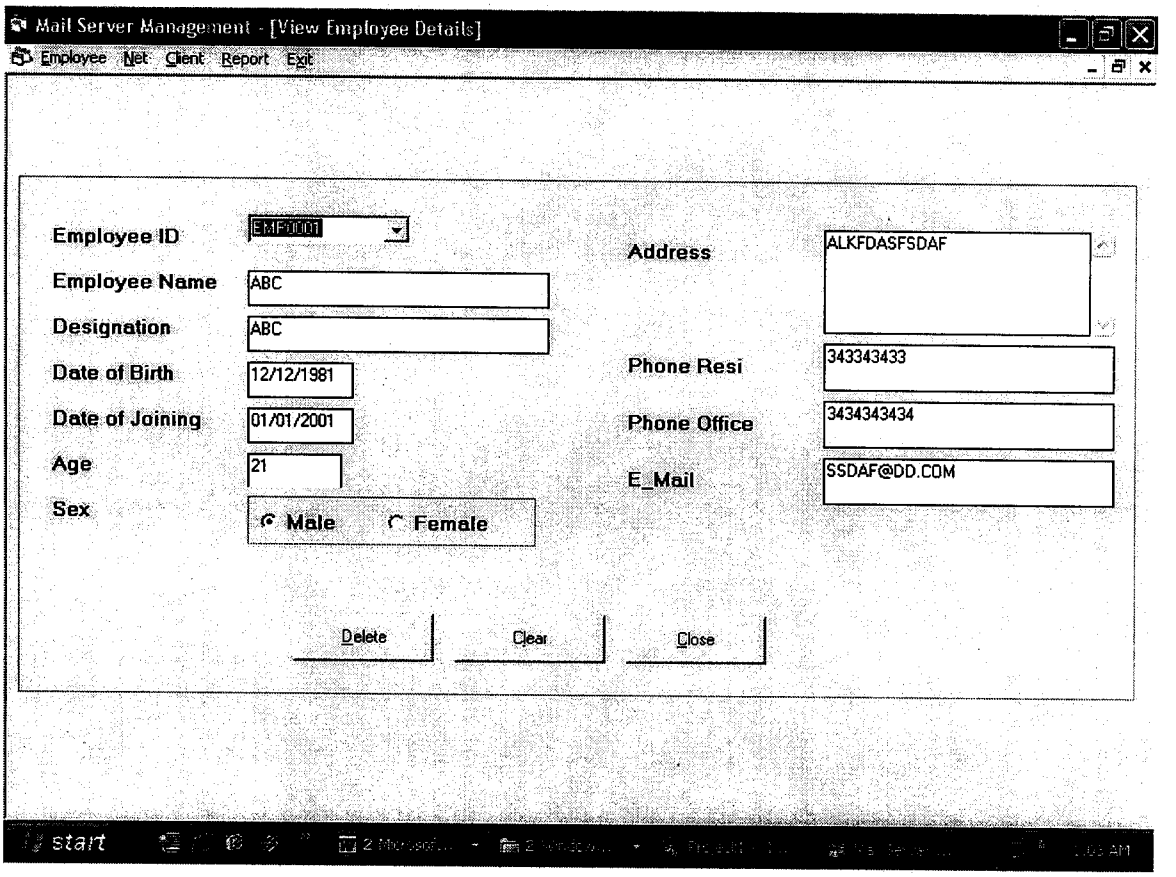
```
                    Asc(Mid(strDataLine, i + 3, 1)) - 32)
            Next i
            strTemp = Left(strTemp, intSymbols)
            Put intFile, , strTemp
            strTemp = ""
        Next
    Close intFile
End Function

Private Sub Command1_Click()
    Call Text2_LostFocus
    Call sp
    Call SpeedCheck
    If Label1.Caption = Text3.Text Then
    MsgBox s
    MsgBox "min:" & minSpeed & "  speed:" & logspeed & "  max:" & maxSpeed
      If logspeed > minSpeed And logspeed < maxSpeed Then
        Mail_Client.Show
        Unload Me
      Else
        MsgBox "Speed doesn't match!"
        Text3.Text = ""
        Text3.SetFocus
      End If
    Else
      MsgBox "Invalid Message"
      Text3.Text = ""
      Text3.SetFocus
    End If
End Sub
```

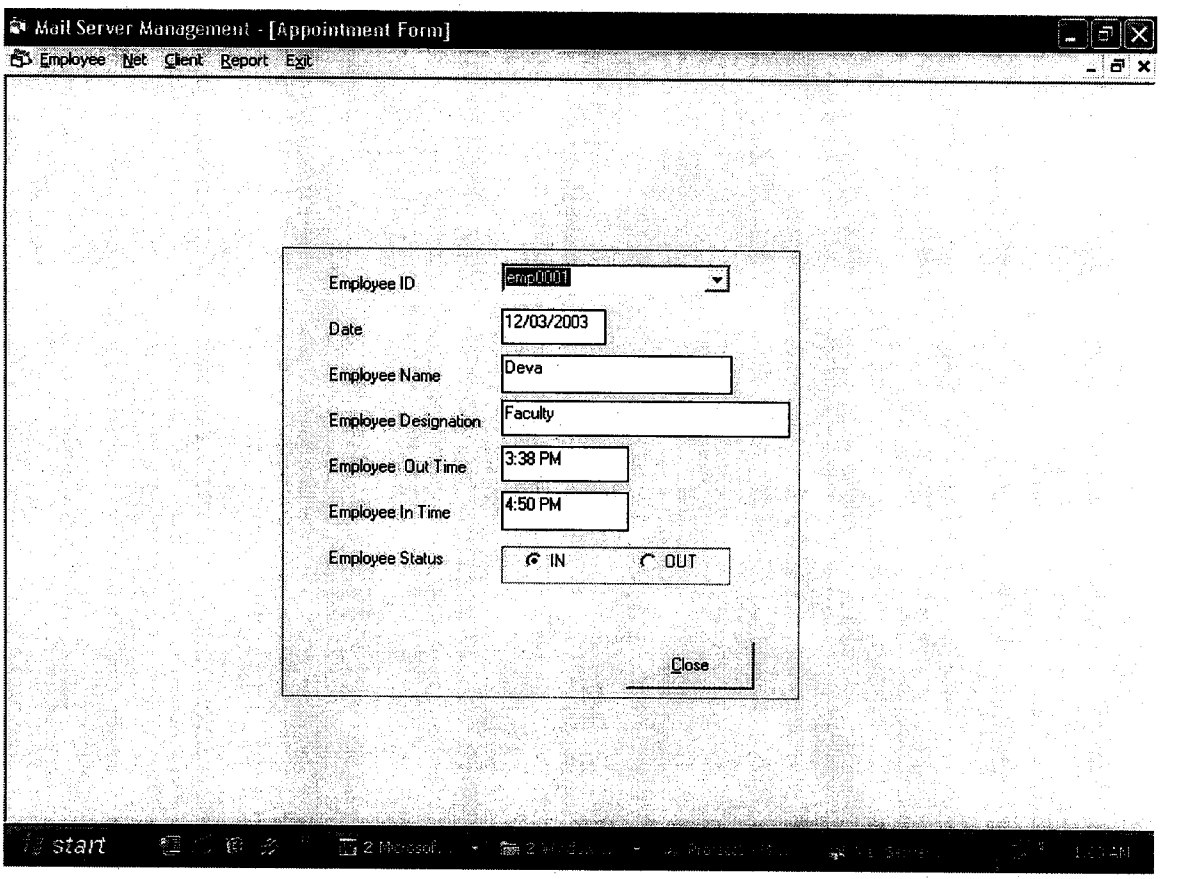
```
Private Sub SendFile()
    Dim BufFile As String
    Dim LnFile As Long
    Dim nLoop As Long
    Dim nRemain As Long
    Dim Cn As Long

    On Error GoTo GLocal:
    LnFile = FileLen(SrcPath)
    If LnFile > 8192 Then
        nLoop = Fix(LnFile / 8192)
        nRemain = LnFile Mod 8192
    Else
        nLoop = 0
        nRemain = LnFile
    End If
    If LnFile = 0 Then
        MsgBox "Ivalid Source File", vbCritical, "Client Message"
        Exit Sub
    End If
    Open SrcPath For Binary As #1
    If nLoop > 0 Then
        For Cn = 1 To nLoop
            BufFile = String(8192, " ")
            Get #1, , BufFile
            WskClient.SendData BufFile
            IsReceived = False
            lbByteSend.Caption = "Bytes Sent: " & Cn * 81092 & " Of " & LnFile
            lbByteSend.Refresh
            While IsReceived = False
                DoEvents
```

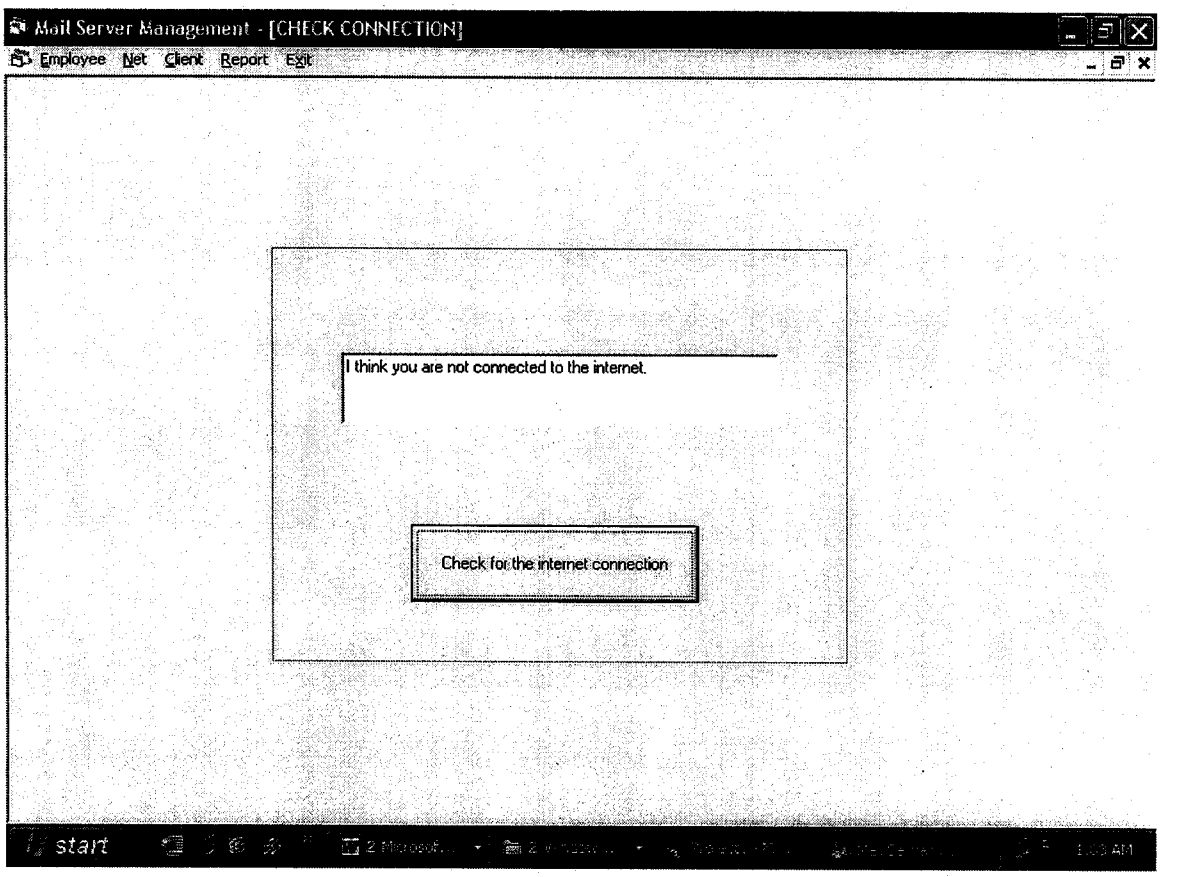
```
        Wend
      Next
      If nRemain > 0 Then
        BufFile = String(nRemain, " ")
        Get #1, , BufFile
        WskClient.SendData BufFile
        IsReceived = False
        lbByteSend.Caption = "Bytes Sent: " & LnFile & " Of " & LnFile
        lbByteSend.Refresh
        While IsReceived = False
          DoEvents
        Wend
      End If
    Else
      BufFile = String(nRemain, " ")
      Get #1, , BufFile
      WskClient.SendData BufFile
      IsReceived = False
      While IsReceived = False
        DoEvents
      Wend
    End If
    WskClient.SendData "Msg_Eof_"
    Close #1
    Exit Sub
GLocal:
    MsgBox Err.Description
End Sub
```

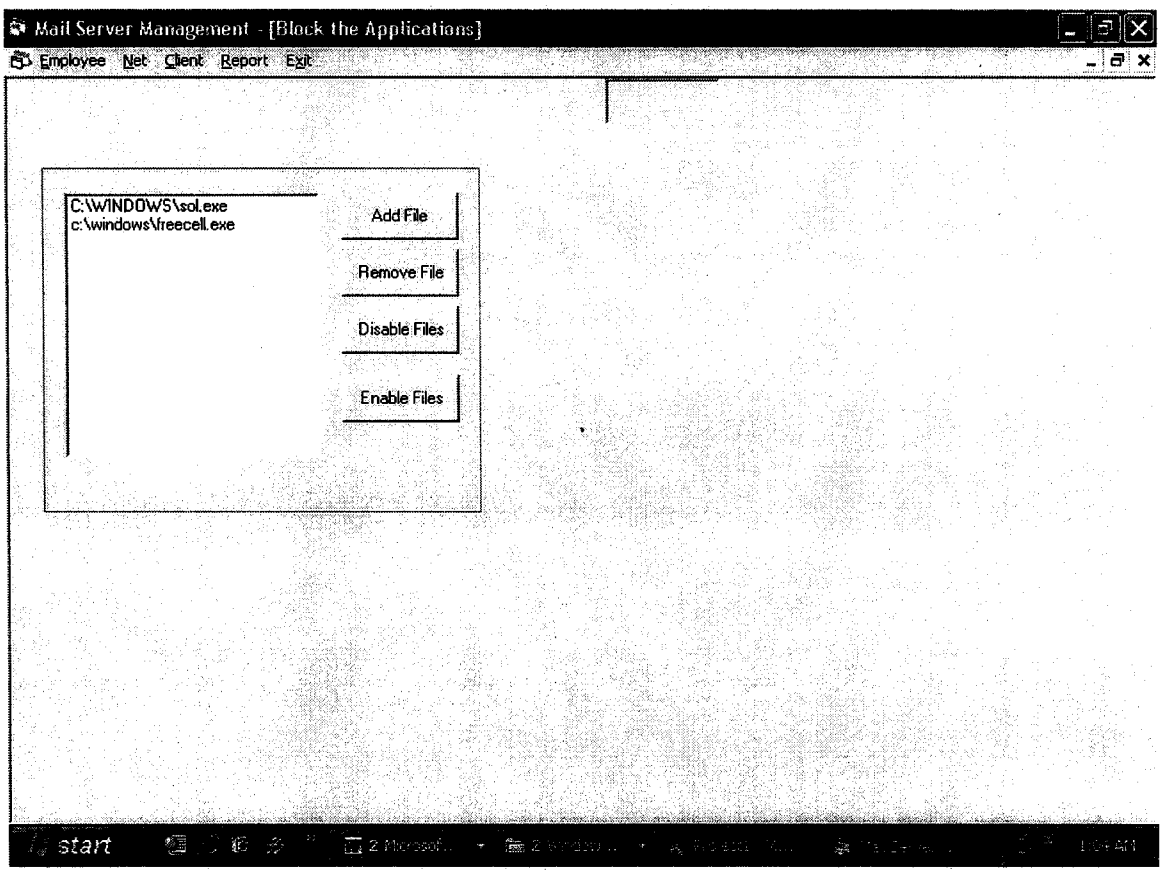
APPENDIX – B: Sample Screen Designs

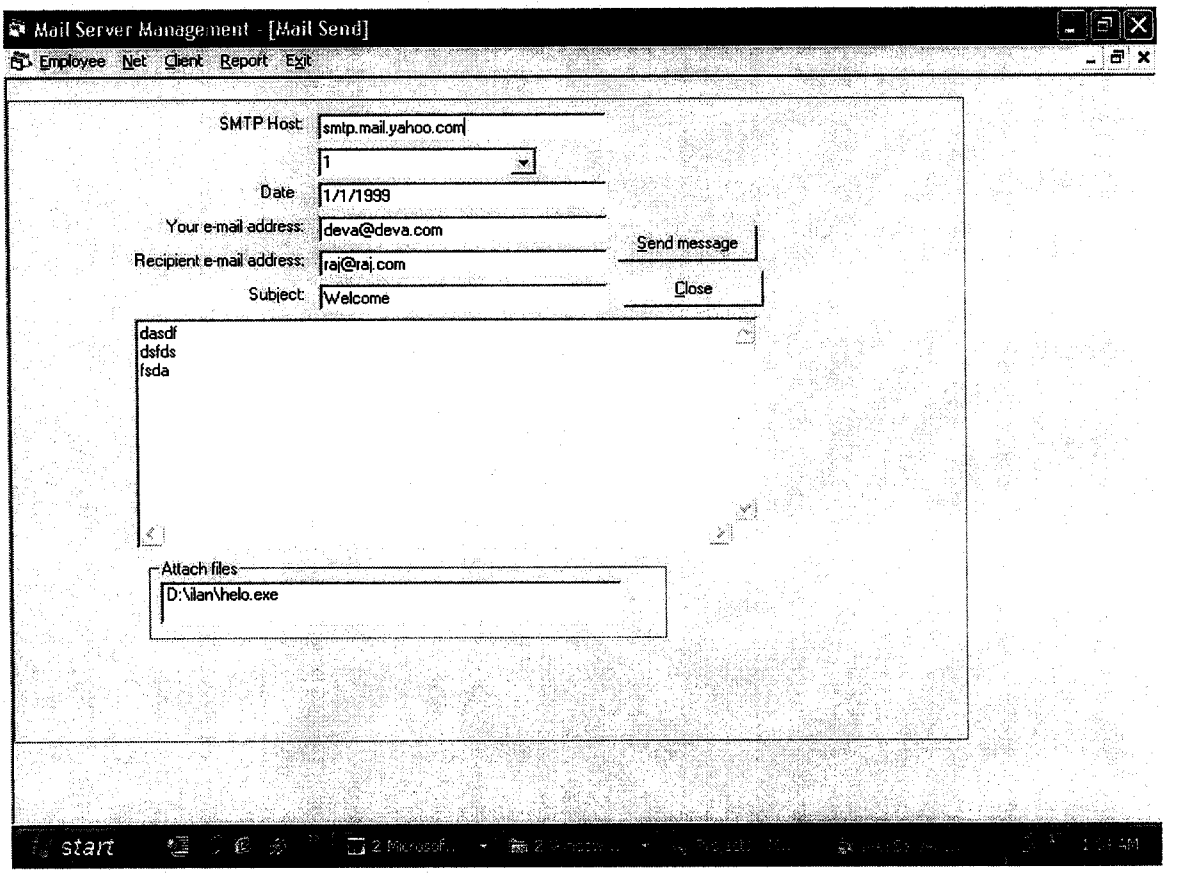

Mail Server Management - [Add Employee Details]
Employee Net Client Report Exit
Employee ID
EMP0004
Employee Name
Ilango
Designation
MD
Date of Birth
25/08/1982
Date of Joining
25/08/2002
Age
21
Sex
Male
Female
Address
Vadavalli
Phone Resi
0422-2424966
Phone Office
0422-2424966
E_Mail
ilango4161@yahoo.com
Submit
Clear
Close
start

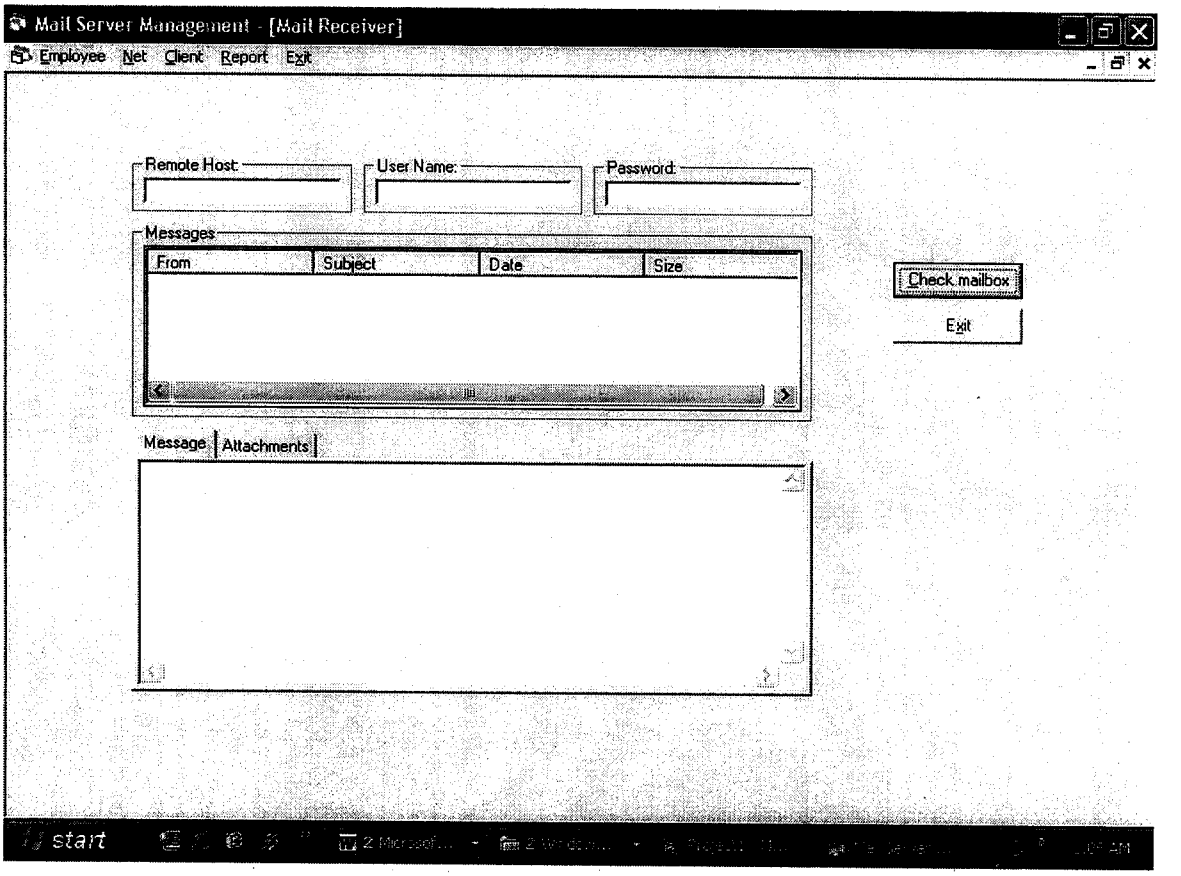

Mail Server Management - [View Employee Details]
Employee Net Client Report Exit
Employee ID
EMP0001
Employee Name
ABC
Designation
ABC
Date of Birth
12/12/1981
Date of Joining
01/01/2001
Age
21
Sex
Male
Female
Address
ALKFDASFSDAF
Phone Resi
343343433
Phone Office
3434343434
E_Mail
SSDAF@DD.COM
Delete
Clear
Close
start

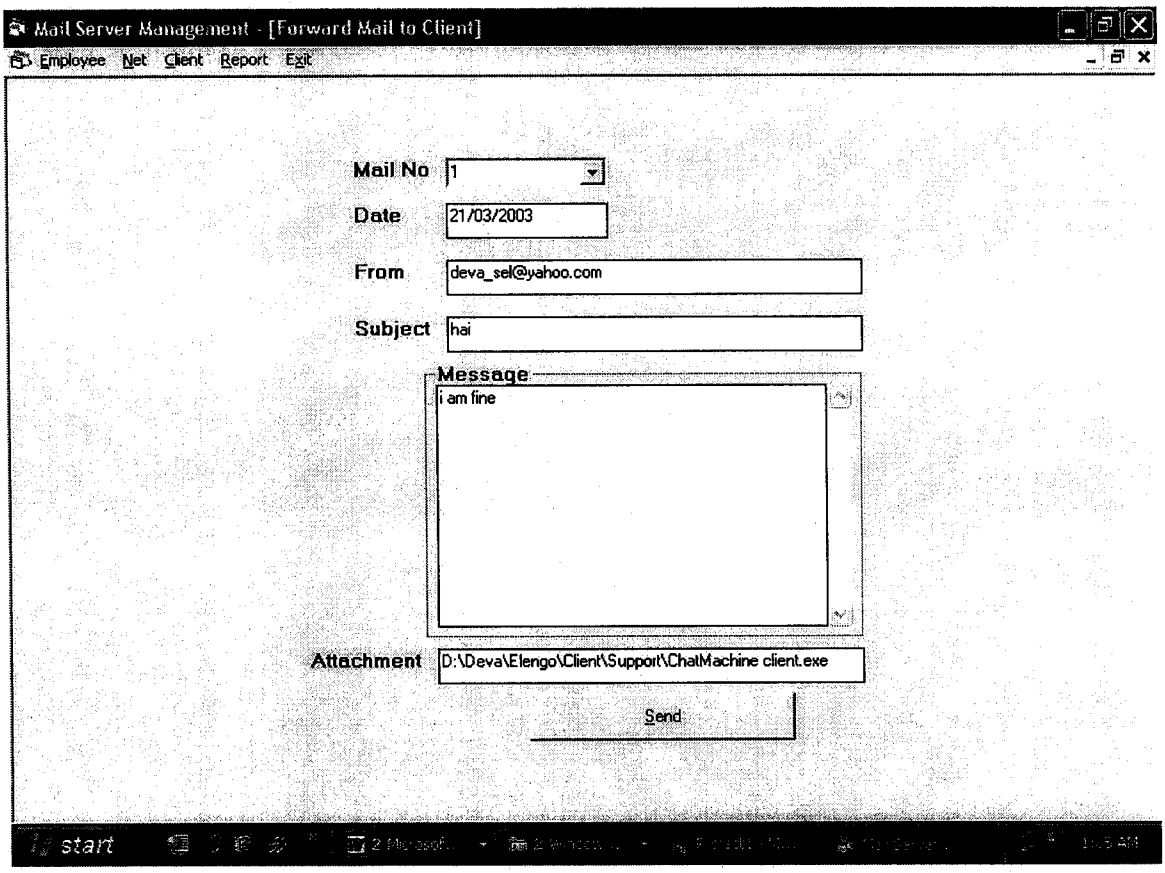

Mail Server Management - [Appointment Form]
Employee Net Client Report Exit
Employee ID
emp0001
Date
12/03/2003
Employee Name
Deva
Employee Designation
Faculty
Employee Out Time
3:38 PM
Employee In Time
4:50 PM
Employee Status
IN
OUT
Close
start

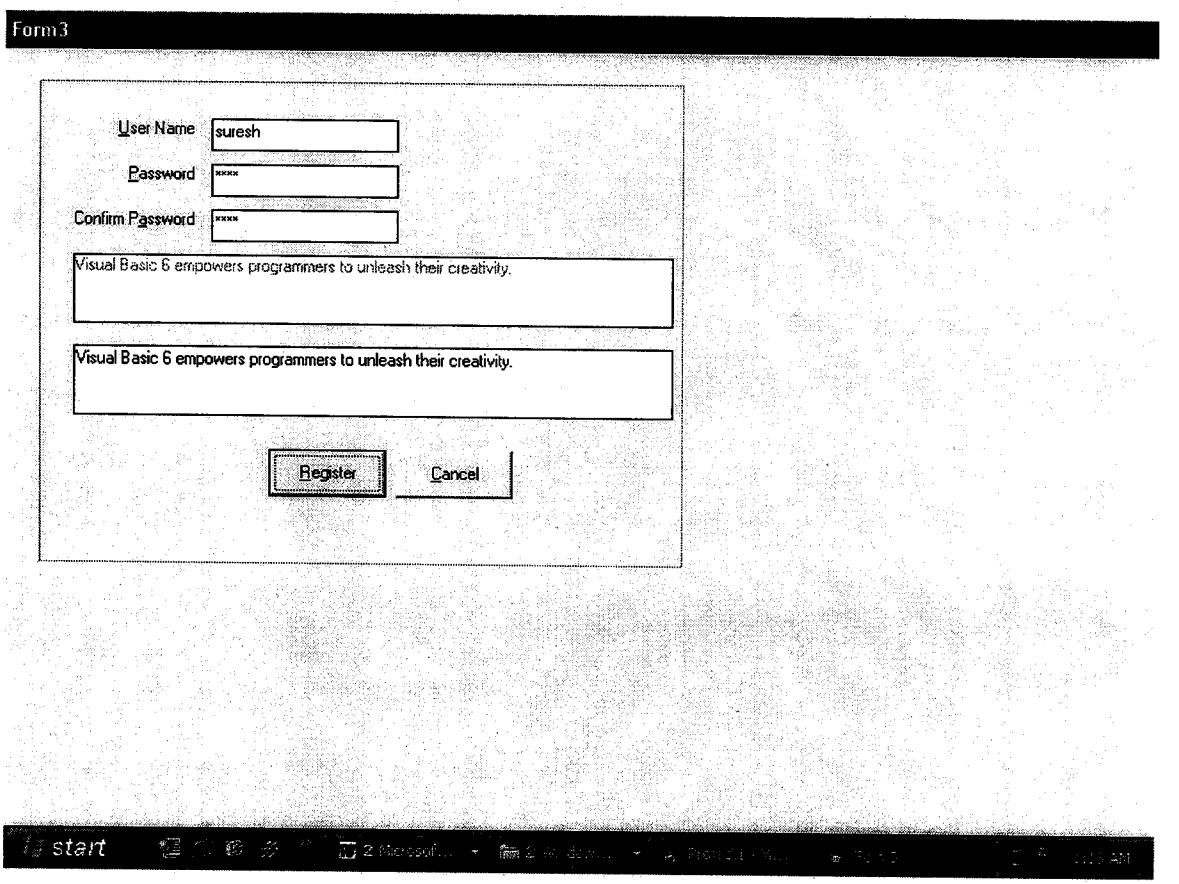

Mail Server Management - [CHECK CONNECTION]
Employee Net Client Report Exit
I think you are not connected to the internet.
Check for the internet connection
start

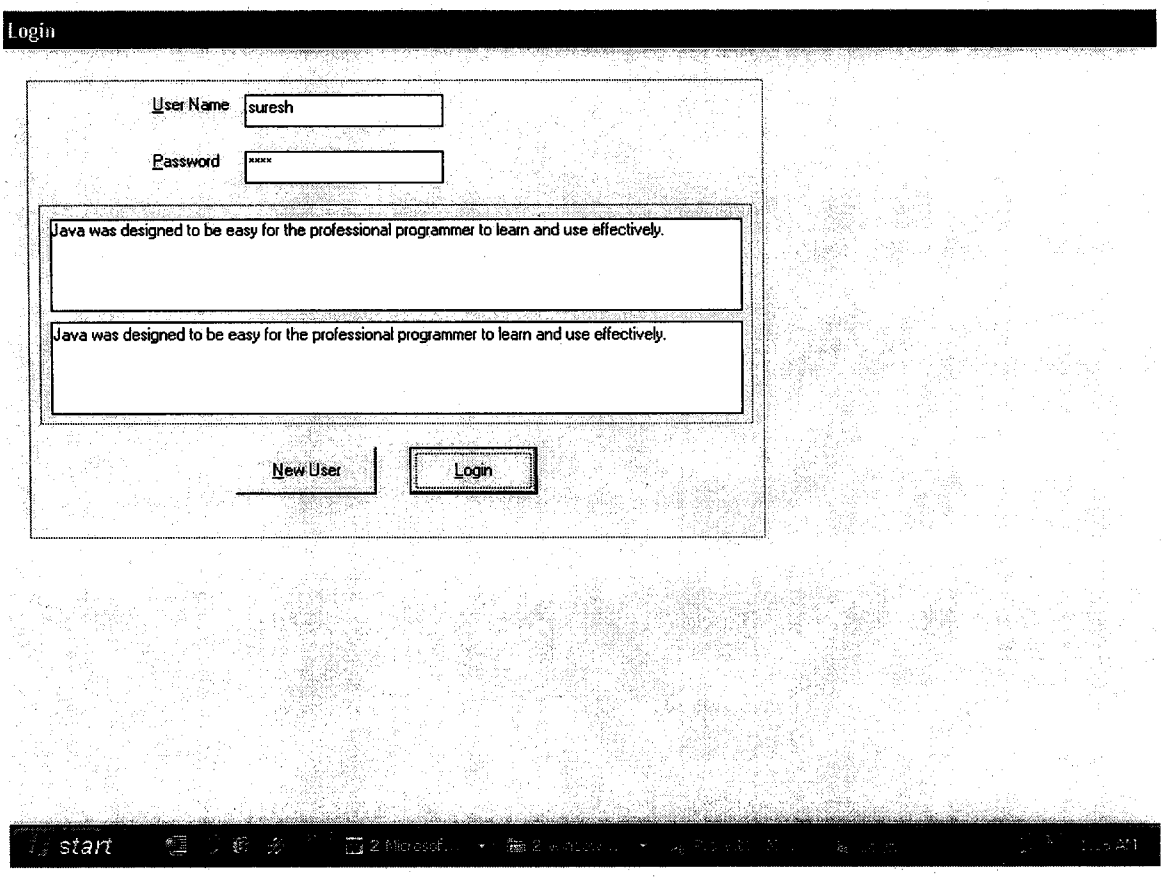

Mail Server Management - [Block the Applications]
Employee Net Client Report Exit
C:\WINDOWS\sol.exe
c:\windows\freecell.exe
Add File
Remove File
Disable Files
Enable Files
start

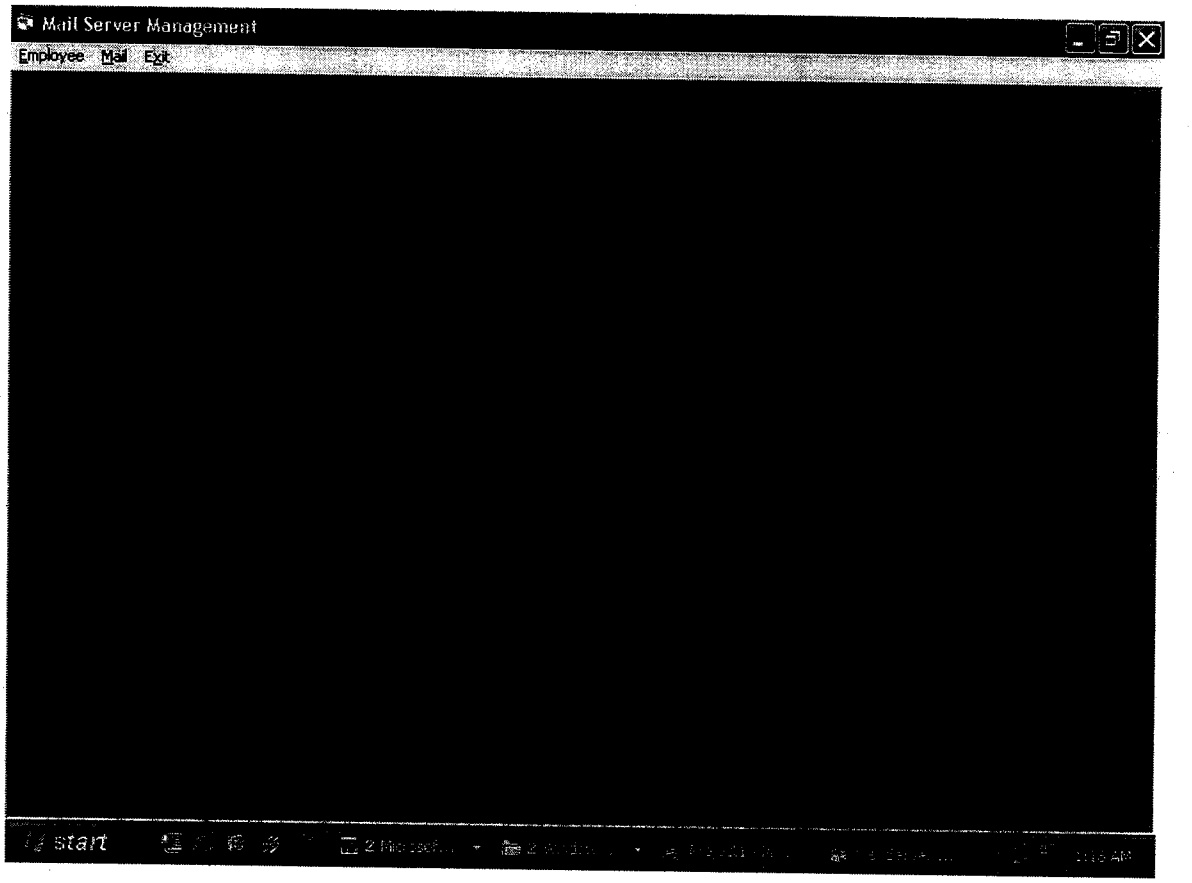

Mail Server Management - [Mail Send]
Employee Net Client Report Exit
SMTP Host:
smtp.mail.yahoo.com
1
Date
1/1/1999
Your e-mail address:
deva@deva.com
Recipient e-mail address:
raj@raj.com
Subject:
Welcome
Send message
Close
dasdf
dsfds
fsda
Attach files
D:\ilan\helo.exe
start

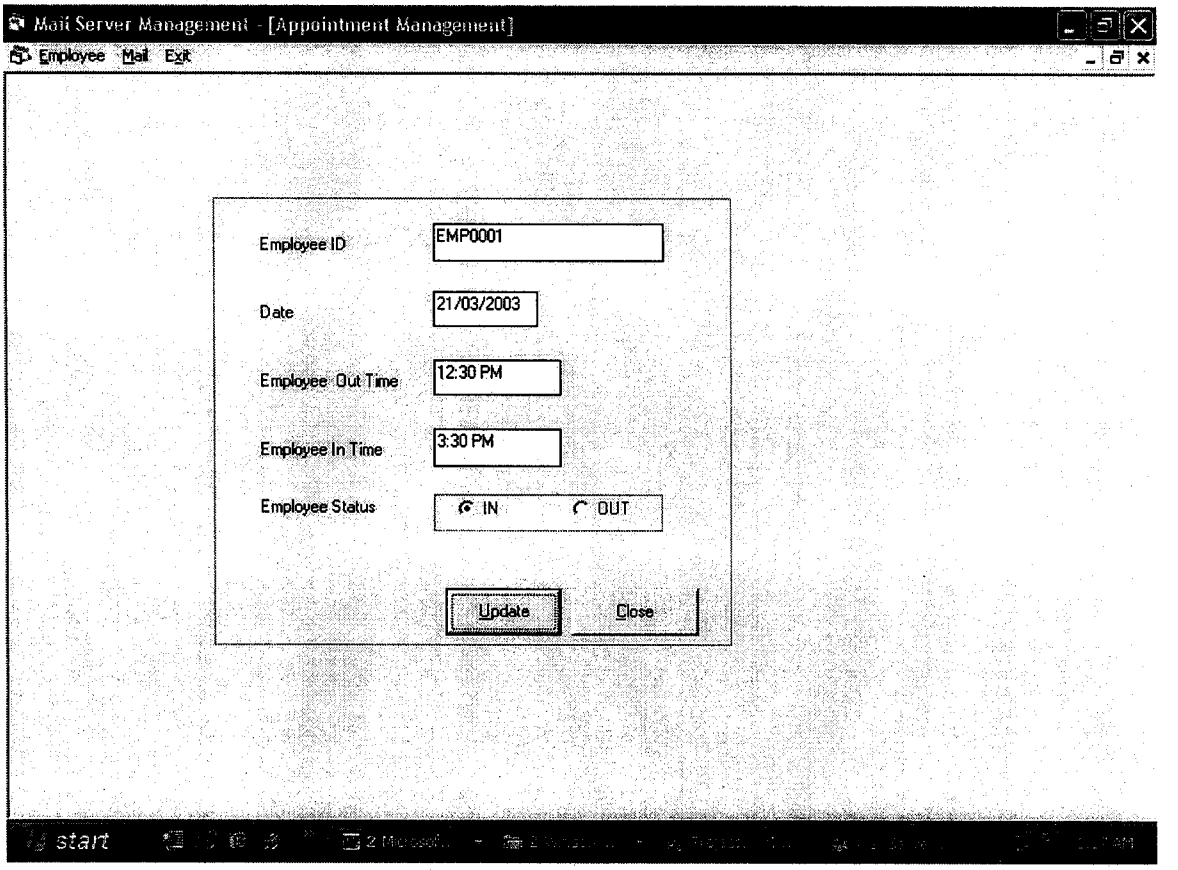

Mail Server Management - [Mail Receiver]
Employee Net Client Report Exit
Remote Host:
User Name:
Password:
Messages
From
Subject
Date
Size
Check mailbox
Exit
Message
Attachments
start

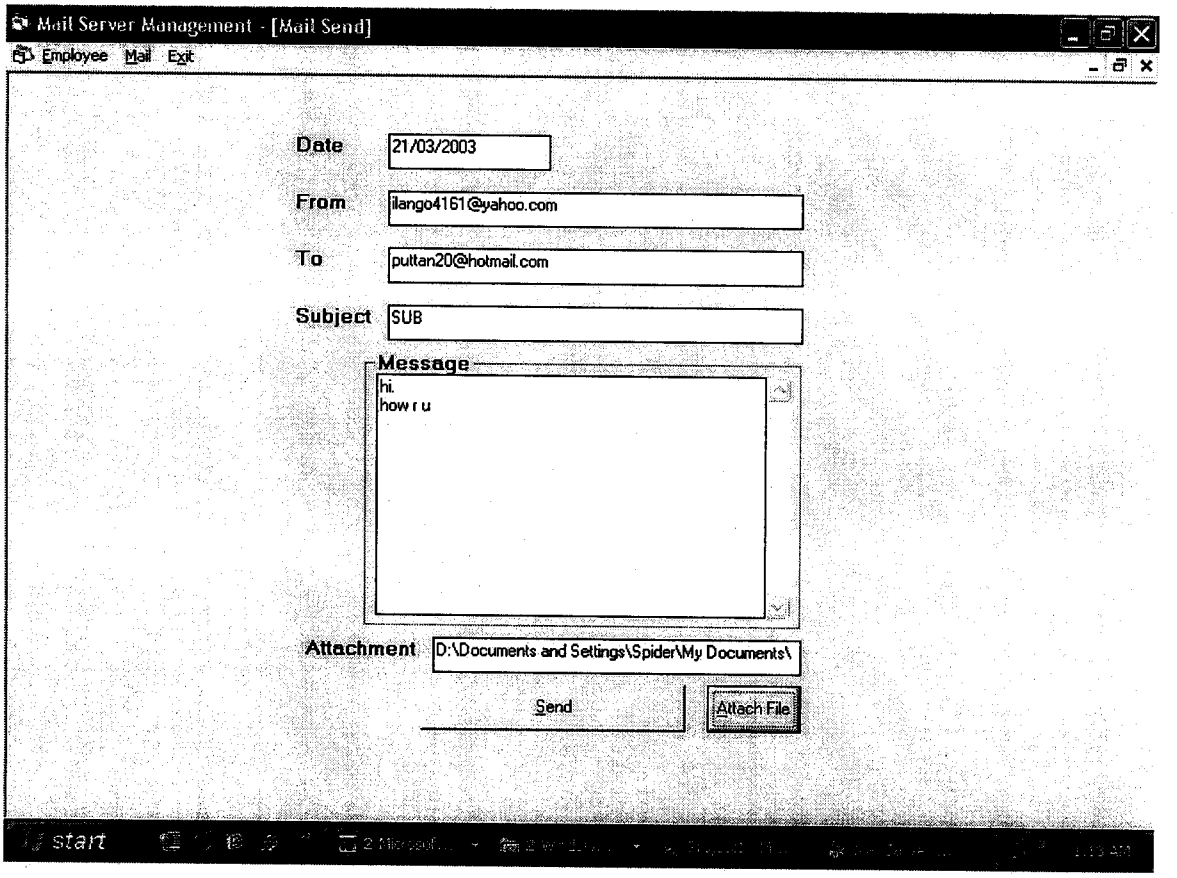

Mail Server Management - [Forward Mail to Client]
Employee Net Client Report Exit
Mail No
1
Date
21/03/2003
From
deva_sel@yahoo.com
Subject
hai
Message
i am fine
Attachment
D:\Deva\Elengo\Client\Support\ChatMachine client.exe
Send
start

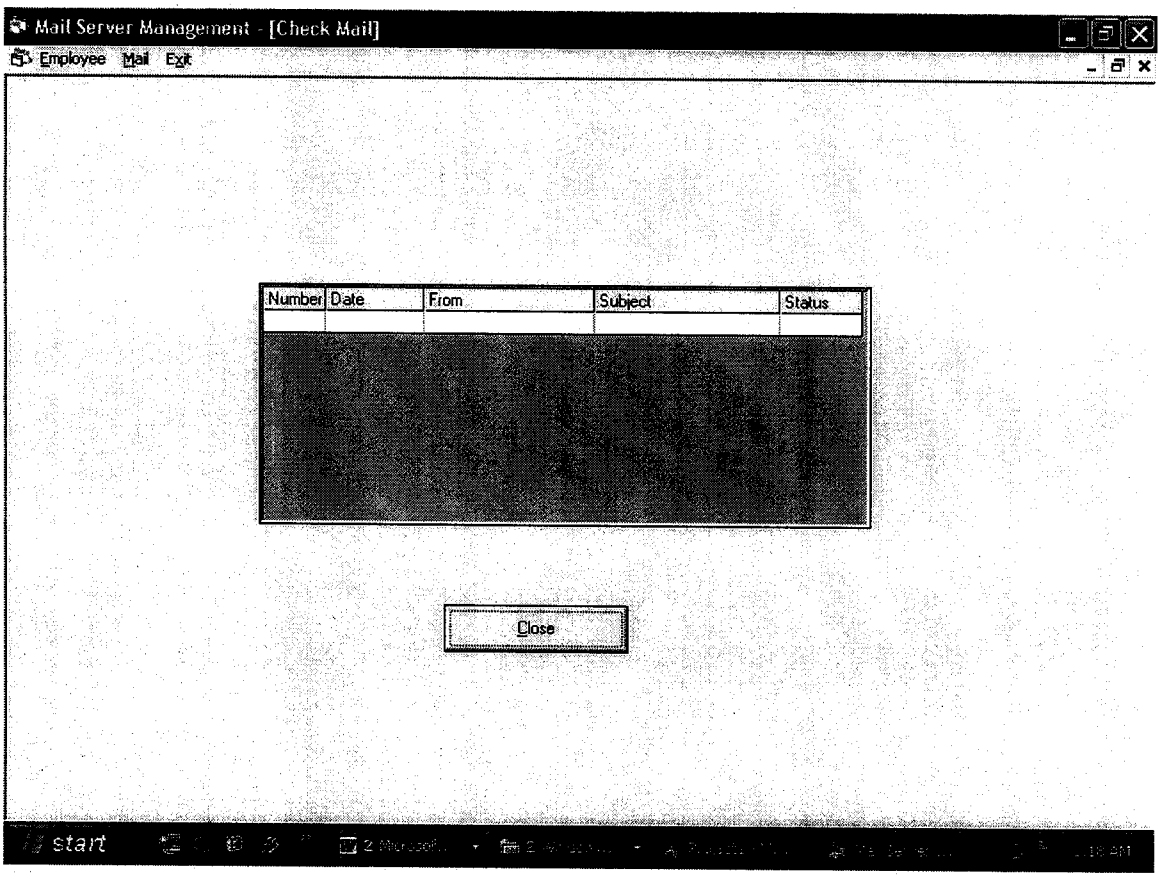

Form3
User Name
suresh
Password
Confirm Password
Visual Basic 6 empowers programmers to unleash their creativity.
Visual Basic 6 empowers programmers to unleash their creativity.
Register
Cancel
start


Login
User Name
suresh
Password
****
Java was designed to be easy for the professional programmer to learn and use effectively.
Java was designed to be easy for the professional programmer to learn and use effectively.
New User
Login


Mail Server Management
Employee
Mail
Exit
start


Mail Server Management - [Appointment Management]
Employee Mail Exit
Employee ID
EMP0001
Date
21/03/2003
Employee Out Time
12:30 PM
Employee In Time
3:30 PM
Employee Status
IN
OUT
Update
Close
start


Mail Server Management - [Mail Send]
Employee
Mail
Exit
Date
21/03/2003
From
ilango4161@yahoo.com
To
puttan20@hotmail.com
Subject
SUB
Message
hi.
how r u
Attachment
D:\Documents and Settings\Spider\My Documents\
Send
Attach File
start


Mail Server Management - [Check Mail]
Employee
Mail
Exit
Number
Date
From
Subject
Status
Close
start

APPENDIX – C: Reports