KIIT UNIVERSITY
Declared U/S 3 of UGC Act 1956

Platform Independent Web Based E Mail System

Web-Based Application Report

Submitted by

AMIR ANSARI (1005015)

Submitted in accordance with the requirements for the degree

Of

Bachelor of Technology

Under the the guidance of
Mr.Subhasis Das
School of Computer Engineering
KIIT University

DECEMBER– 2013

CERTIFICATE

This is certifying that Mr. Amir Ansari, B.TECH in Computer Science and Engineering student of KIIT UNIVERSITY has submitted project under my guidance.

Mr.Subhasis Das
School of Computer Engineering
KIIT University

ACKNOWLEDGEMENT

After having completed my project report, there remains only the pleasant task of acknowledging the help I received in writing my reports. I really feel an immense pleasure to acknowledge all those people who had constantly support me during the project making & helped me a lot in writing my project report. It’s almost certain that without their valuable & timely suggestions, it would have taken me years to complete.

DECLARATION BY CANDIDATES

I hereby declare that the work entitled "Platform Independent Web Based E Mail System **"** is an authentic work carried out by me at **"KIIT UNIVERSITY "**under the guidance of Mr. **Subhasis Dash** , for the fulfillment for the award of the "B.Tech in Computer Science and Engineering " and this has not been submitted anywhere else for the award of any other degree.

1.INTRODUCTION

Information is one of the most valuable things of the current information society. Fast access to needed information is critical in business at the present time. Companies have to spend huge amount of money for transferring information using various ways to their employees in field. Communication can be done in numerous ways; the way depends on the distance between the sender and the receiver of the information. Also each way of communication has its own cost. Two persons close to each other communicate by their own voice or by exchange of some storage medium. When the information has to be passed to a longer distance, there are multiple options for it, as we can pass information using:

- **Postal system**
- **Telephones**
- **E-Mails**
- **Broadcasting etc.**

Generally we use telephones, letters & E-Mails only, from these postal system is not so reliable and too slow, so use of postal system is decreasing, telephones are the easiest one, but cannot be used for large amount of data and cost a lot, and the last the E-Mail, this is cheap, reliable and also very fast, and the best thing about them is that nowadays you don’t even need a computer for sending and receiving E-Mail, it can even be done through mobile. It can also transfer large amount of data as well as computer data files and programs.

So some of the features of e-mail which make it one of the most used services of the internet are:

- **Speed.**
- **Reliability.**
- **Low-Cost.**
- **Ease of use.**

Due to the above mentioned features e-mail is gaining more and more acceptance with each passing day. Individuals as well as businesses both type of users have started using e-mail for their communication.

After a good amount of study of the web-based e-mail system, the following objectives in order to meet the demand of a web-based e-mail system were decided. The title was decided as **Platform Independent Web-Based E-Mail System** and will be developed using MVC architecture of JSP. There are some main objectives of this project which are given below:

- The system must not use any commercial software for which the company has to buy any software licenses for using the system.
- Sending text messages.
- Receiving text messages.
- Automatically entering of e-mail addresses when selecting nick name of a person whose name is in address book.
- Automatically adding new e-mail addresses to which mail is send in address book.
- Delete the mail.
- New user can create E-mail account.
- Password Recovery system using a Secret Question and other details.
- Adding new record in address book.
- Deleting any record from address book.
- Updating any record in the address book.
- Change user password.
- Change user details
- Logout from current login.

A small example can help us easily understand the need of e-mail. Let's say you have a small business with sales representatives working around the country. How do you keep in contact without running up a huge phone bill? Or what about keeping in touch with far-flung family members? E-mail is the easiest and cheapest way. It's no wonder e-mail has become the most popular service used on the Internet.

Due to so many inherent advantages e-mail has become one of the most used forms of communication by most of the companies, and the rest are too starting to use e-mail for most of their communications.

WHY WEB-BASED E-MAIL

Web based mailing system further enhances the features of simple email system by adding some features which are as following:

UNIVERSAL ACCESSIBILITY

Web based e-mail is accessible from any place in this world there is no need of your own computer. You can also access it using your mobile phone. All you have to do is open the website of the provider and use your login and password and start reading and writing mail. This universal accessibility is one of the biggest advantages of web-mail.

ADDRESS BOOK

Web based e-mail also stores your contacts on the web so all your important contacts are accessible from any place in this world. This is of great help for a person who checks his mail from different places.

NO NEED OF MAIL SOFTWARE

E-Mail normally requires an e-mail client installed on the computer to receive and send email but web-based e-mail requires only a computer connected to the internet having a web browser installed, and every computer has that installed.

NO NEED OF LOCAL STORAGE

E-Mail clients normally require space on the computer used to receive e-mails and they store past e-mail on the local computer.

2. OBJECTIVES

After a good amount of study of the web-based e-mail system, the following objectives in order to meet the demand of a web-based e-mail system were decided. The title was decided as **Platform Independent Web-Based E-Mail System** and will be developed using MVC architecture of JSP. There are some main objectives of this project which are given below:

3.0 SYSTEM ANALYSIS

3.1 IDENTIFICATION OF NEED OF E-MAIL

A small example can help us easily understand the need of e-mail. Let's say you have a small business with sales representatives working around the country. How do you keep in contact without running up a huge phone bill? Or what about keeping in touch with far-flung family members? E-mail is the easiest and cheapest way. It's no wonder e-mail has become the most popular service used on the Internet.

Due to so many inherent advantages e-mail has become one of the most used forms of communication by most of the companies, and the rest are too starting to use e-mail for most of their communications.

3.2 WHY WEB-BASED E-MAIL

Web based mailing system further enhances the features of simple email system by adding some features which are as following:

3.2.1 UNIVERSAL ACCESSIBILITY

Web based e-mail is accessible from any place in this world there is no need of your own computer. You can also access it using your mobile phone. All you have to do is open the website of the provider and use your login and password and start reading and writing mail. This universal accessibility is one of the biggest advantages of web-mail.

3.2.2 ADDRESS BOOK

Web based e-mail also stores your contacts on the web so all your important contacts are accessible from any place in this world. This is of great help for a person who checks his mail from different places.

3.2.3 NO NEED OF MAIL SOFTWARE

E-Mail normally requires an e-mail client installed on the computer to receive and send email but web-based e-mail requires only a computer connected to the internet having a web browser installed, and every computer has that installed.

3.2.4 NO NEED OF LOCAL STORAGE

E-Mail clients normally require space on the computer used to receive e-mails and they store past e-mail on the local computer. This can be a problem when you are constantly on the move and you do not have access to your computer. Web based e-mail stores all your e-mails on the internet until you delete them so you can even check your past messages from anywhere.

3.3 PRELIMINARY INVESTIGATION

To know what the real problem is one of the biggest milestone in its solution so for this a study was initiated so as to understand the working of a web-based e-mail system. With the knowledge gained from the study the following components were identified which constitute a web based e-mail system.

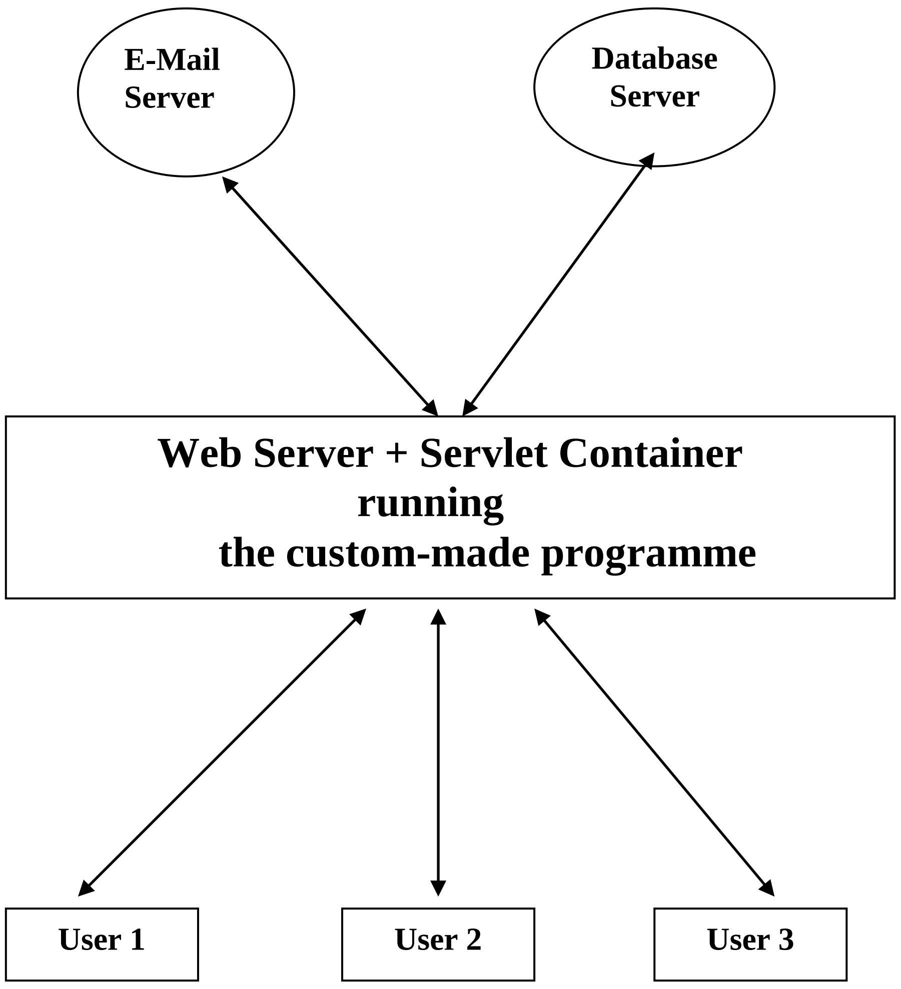
- ❖ **E-Mail Server.**
- ❖ **Database Server.**
- ❖ **HTTP Server.**
- ❖ **Servlet Container.**
- ❖ **Few Libraries.**

Two protocols were also identified that has to be used for transferring and receiving of e-mail form the server which are as follows:

- ❖ **SMTP = Simple Mail Transfer Protocol.**
- ❖ **POP = Post Office Protocol.**

After doing the preliminary study the architecture of the web-based e-mail system was also identified. The custom made application runs on the HTTP Server which also has a servlet container and this application contacts with the Users, E-mail Server and the Database Server. A pictorial representation is given on the next page.

3.4 PICTORIAL REPRESENTATION OF ARCHITECTURE


E-Mail Server
Database Server
Web Server + Servlet Container running the custom-made programme
User 1
User 2
User 3

FEASIBILITY STUDY

4.1 FEASIBILITY

Once the scope for the e-mail project was defined, the next most important question to be asked was is this project feasible. As I haven’t done a project relating to e-mail before the answer was not easy. A good amount of investigation was needed on various points to understand the feasibility of this project. Feasibility study was done on mainly three points which are as follows:

- **Technical Feasibility.**
- **Economical Feasibility.**
- **Operational Feasibility.**

Of these Economical and Operational Feasibility weren't much important as this project is made only for educational purpose not for any client. Only technical feasibility was studied in detail.

4.2 TECHNICAL FEASIBILITY

Technical Feasibility is very important for any project. It is important to decide in advance that all the equipment and tools needed by the project must be available. This project uses very basic hardware present in any computer so it will be easily available everywhere. The software tools used by this project are also easily available so there wasn't any difficulty.

The only problem was in deciding was that whether this project can be made by the technical skills that I have but after a lot of study and a little help from guide this project looked technically feasible.

4.3 ECONOMICAL FEASIBILITY

As the project is made for pure educational purpose there wasn't much need of testing its economical feasibility. Yet I did some test for the cost analysis of the software tools and hardware required for this project. On the hardware front the project needed was the basic hardware that is present in every computer. A computer was freely available so it didn’t cost anything.On the software front all the tools and software needed for this project are freely available on the internet so there wasn’t much cost associated with them except for the cost associated in downloading them. Lastly as the project is going to be made by me so there is no cost associated in developing the project. So this project is economically feasible.

4.4 OPERATIONAL FEASIBILITY

Any software how much useful it is fails to make any impact if it is not easily usable. As e-mail is used by persons who are not very comfortable with computers so a great care was taken to make the web interface as simple as possible, so that it is easily usable by anyone. It is designed using the layout as most other e-mail services provide, so it will be very easy to use for anyone who has used e-mail before.

Only at the time of installation a trained person will be required and all other maintenance can be done by any computer user. The software doesn’t need much care once started it doesn’t needs much human intervention.

SOFTWARE ENGINEERING PARADIGM APPLIED

5.1 SOFTWARE ENGINEERING PARADIGM USED

For the systematic development of a project some software engineering paradigm is selected by the software engineer before the beginning of the project. The paradigm is chosen based on the nature of project, time, tools and methods to be used in the development of the project. In my web based e-mail system I also had to decide a paradigm to use. There were few properties and requirements of the project that helped me in deciding the paradigm to be used:

- ❖ Deadline for the final product 20 September 05.
- ❖ Ease to identify the requirements, inputs and outputs as model systems were already present to understand the working.
- ❖ Requirement to develop an intermediate version with core functionality before going on examination break from 1 June 05.

The above three points especially the last one strongly suggested in using the **Incremental Model** of software engineering. Also the incremental model is a good choice when there is insufficient number of developers. In my project I was the only person in charge of everything so it was a good option for me. After deciding the Incremental model it was easy to decide the number of increments that I should make for this project. I decided for making two versions of the project:

1.Version 1.0 - This version must be completed before going on the examination break and should have all the core functionality of composing, sending and receiving messages. It must also have the option of deleting mail.
2.Version 2.0 – This final version must be complete in all respects before 20 September 05 and must have all the additional functionality of address book, password retrieval as well as other optional components.

In Incremental model normally; the increments may have there some stages of development, in parallel like the first increment may be in coding stage while the second increment might be in analysis stage. But as in my case as I was the only person to do everything; I completed the first version before starting the second version. The pictorial representation of the programming paradigm used is given on the next page.

SOFTWARE & HARDWARE REQUIREMENT SPECIFICATION

6 HARDWARE/SOFTWARE REQUIREMENT SPECIFICATION

This project has to be quite flexible in terms of requirements needed. It should use the existing machines and must not have any special requirement. It should allow all of the existing machines to be able to use the E-Mail System without any (or with very less) changes. Also the project must use the existing software, or free software that does not have any licensing issue to keep the cost less. The requirements are as follows:

6.1 SERVER-SIDE REQUIREMENTS

6.1.1 Hardware Requirements:

❖ **Processor**	**:**	Pentium III 800 MHz or above.
❖ **RAM**	**:**	128 Mb or above.
❖ **Hard Disc**	**:**	20 GB or above.

6.1.2 Software Requirements:

❖ **Operating System**	**:**	Windows 98/ME/2000/XP Linux. With JVM (Java Virtual Machine) installed.
❖ **Web Server**	**:**	Apache Tomcat (for Win/Linux).
❖ **Database Server**	**:**	MySQL (for Win/Linux), MyODBC DSN Creator

❖ **Mail Server**	**:**	MailEnable Standard (for Win only) Apache James (for Win/Linux) Or Any Other Mail Server, running SMTP at port 25 and POP at port 110.
❖ **Libraries**	**:**	Activation.jar, Mail.jar
❖ **Web Browser**	**:**	Internet Explorer, Netscape, Mozilla Any web browser which is JVM (Java Virtual Machine) compatible.

6.2 CLIENT-SIDE REQUIREMENTS

6.2.1 Hardware Requirements:

❖ **Processor**	**:**	Pentium II 350 MHz or above.
❖ **RAM**	**:**	32 Mb or above.
❖ **Hard Disc**	**:**	2 GB or above.

6.2.2 Software Requirements:

❖ **Operating System**	**:**	Windows 98/ME/2000/XP Linux etc. Any Operating System which is JVM (Java Virtual Machine) compatible.

❖ **Web Browser**	:	Internet Explorer, Netscape, Mozilla Any web browser which is JVM (Java Virtual Machine) compatible.

TOOLS AND LANGUAGES USED AND THEIR FEATURES

7 TOOLS/PLATFORM, LANGUAGES USED

Every project needs some tools and programming language that has to be decided in advance during the analysis. This project also uses quite a number of tools as well as languages which are explained below in detail. This project can work on both Windows and Linux platforms but I am developing and testing only on windows platform:

7.1 TOOLS USED

This project was developed on a Windows XP system and it uses various software tools which are given below along with their use in detail:

❖ **Oracle JDeveloper –** Oracle JDeveloper is Oracle's Java development tool for building, debugging, and deploying Internet applications. It includes Oracle Business Components for Java, a 100% Java, XML-powered application component framework that dramatically simplifies the development, deployment, and customization of multi-tier Java applications for the Internet. JDeveloper was used for the coding all the files and testing them.

❖ **J2SE Development Kit –** Java 2 Platform standard edition includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform. These tools are designed to be used from the command line. J2SE also includes JRE (Java Runtime Environment) that is used for running programs that are written in Java.

- **MySQL –** MySQL has become the most popular open source database and the fastest growing database in the industry. MySQL offers several key advantages:
  - **Reliability and Performance**
  - **Ease of Use and Deployment**
  - **Cross-Platform Support**
  -
- **MyODBC –** It is an add-on for the MySQL database. It is used for creating a DSN on the windows platform for easy connectivity with the database.

- **MailEnable Standard / Apache James –**
  - **MailEnable mail server** software provides a powerful, scalable messaging platform for Microsoft Windows. MailEnable offers stability, unsurpassed flexibility and an extensive feature set which allows to provide cost-effective mail services. Some features of MailEnable are as follows:
    - **Unlimited Users/ Domains**
    - **Eliminate Spam**

  - **Apache James** the Apache Java Enterprise Mail Server is a 100% pure Java SMTP and POP3 Mail server and NNTP News server. James is designed to be a complete and portable enterprise mail engine solution based on currently available open protocols. One of the remarkable features of Apache James is that it can run easily on both Windows and Linux and is completely free to be used by any organization with an unlimited number of users.

❖ **Apache Tomcat –** Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed by Sun under the Java Community Process. Tomcat is developed in an open and participatory environment. It is free open source and works on both Windows and Linux.

7.2 LANGUAGES USED

The project uses the **J2EE platform** and its technologies. Some technologies will be used for a little work and some will be used throughout. The following is a complete list of all technologies and their use in the project in detail:

❖ **JAVA** – It is one of the most powerful Object Oriented Platform Independent Language. Java is used in most of the project in one form or another. Bean that is to be used in the project is a Java application.

❖ **JSP –** Java Server Pages or JSP for short is a server-side technology that takes Java language with its inherent simplicity, and uses it to create highly interactive and flexible web applications. JSP will be used in the project to create most of the web pages that interact with the user.

- **JavaScript –** JavaScript is used for client- side scripting. It enables the web pages to have some programmatic functionality in the browser. JavaScript works with and can manipulate the HTML page in which it is embedded. JavaScript is used in the project to validate the data entered in the web pages by the user before it is send to the server.

- **Servlet –** A Servlet is a Java Program that generates dynamic web content. They are written using the Java Servlet API and are managed by a Servlet container such as Tomcat. The Servlet processes the user request, builds a response, and passes it to the container which it back to the user. I have used a Servlet for making the controller in the web based e-mail system. It processes the user requests and gives appropriate response according to them.

- **HTML –** HTML or Hyper Text Markup Language is used creating web pages. HTML pages are static and do not interact with the user but can be made interactive by adding JSP elements them. Most of the web pages in the project are designed in HTML and after that JSP elements are added to them.

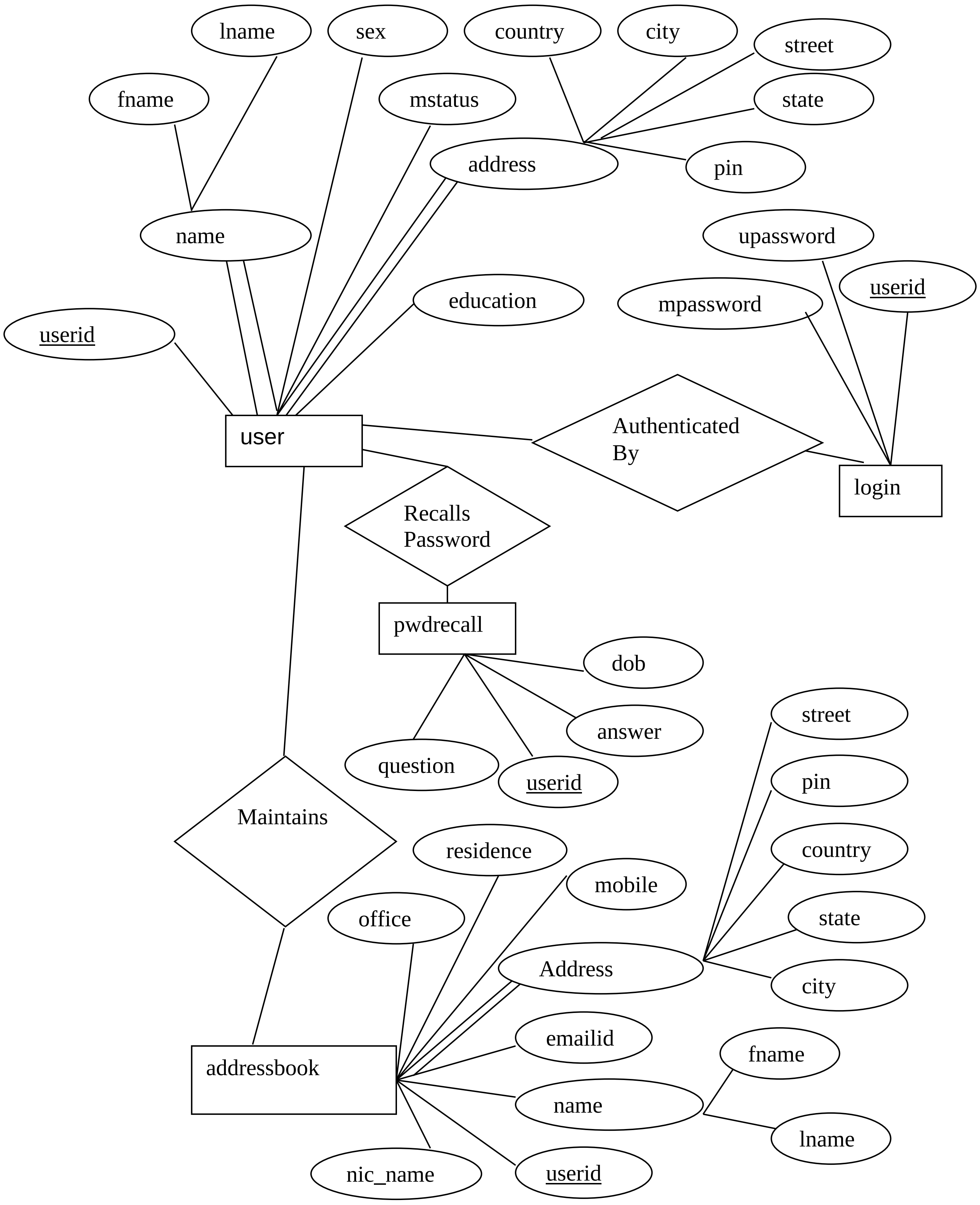
- **XML –** Extensible Markup Language or XML for short has become the de facto standard for data interchange on the Internet. We have not used it directly but a lot of configuration files of Apache Tomcat Server as well as Apache James E-mail server are written in this format.

SYSTEM DESIGN

8.1 ENTITY/RELATIONSHIP DIAGRAM

Entity/Relationship diagram or ER diagram in short are used to graphically represent entities (data objects), their attributes and relationships between objects. To make an ER diagram primary components are identified: data objects, attributes, relationships, and various type indicators. Entities are represented by labelled rectangle. Relationships are indicated with lines that contain a diamond labelled with the relationship connecting objects. The ER Diagram of “web based E-mail system” is given on next page.

8.2 ER DIAGRAM



8.3 DATA DICTIONARY

- **answer** It stores the secret answer to the question asked from the user in the password reminder procedure when the user forgets his password.
- **city** Contains the city name of the address.
- **country** Stores the name of the country in the address.
- **dob** Date of birth of the user. It is used in the password reminder process when the user has forgotten his password.
- **email** Stores the email address of the friend whose contact is in the address book.
- **fname** First name of the person.
- **lname** Last name of the person.
- **mobile** Contains the mobile number.
- **mpassword** Stores the mail server password of the user. This is different from the user password. User has no control over it.
- **mstatus** Contains the marital status of the user.
- **nic_name** Stores the nick name of the person who is in the contact. A user needs only to click on the nick name and the address is automatically entered in the "To:" field of the compose page.
- **occupation** Contains the occupation of the email user.
- **office** Contains the office phone number of the contact in the address book.
- **pin** Stores the Pin No of the address.
- **question** Stores the question that is asked in the password reminder process that is used when the user has forgotten his password.
- **residence** Contains the home phone number of the person who is in the contact list.
- **sex** Stores whether the user is male or female.
- **state** Stores the name of the state in the address.
- **street** Contains the street name of the address.

- **upassword** Stores the user password for logging in to the site. It is different from the mail server password. User has control over this password and can change it when he wants.

8.4 DATA TABLES WITH SCHEMA & KEYS

Database design is the most important part in any project. As most of the other parts of the program depend upon database it has to be done before any other thing. A project with a badly designed database often leads to its failure. So after a thorough study database tables were designed so that there is a good balance between access-time and normalization.

There are a few things that are done against the rule of normalizations for performance enhancement which are as follows.

- "login" table was not needed according to the rules of normalization as the data it contains was related to user details and must be in the user information table called user. But most of the queries to the database will be for userid and upassword so they were moved to other table called login which will be short as compared to user, so fast access will be possible.

- In the same way pwdrecall table was not needed as the data it contains was related to user details and must be in the user information table called user. There was a significant difference of the type of access needed for both tables. Also the data in this pwdtable is very important where as the data of user is not of much use.

So finally it was decided to make four tables in the library database which as follows:

1. **login**
2. **user**
3. **pwdrecall**
4. **addressbook**

TABLE STRUCTURE

1. **login:** This table contains the login information about the user. This table is made short because it will be the most accessed table of the whole database. It contains the following fields:

Field Name	**Data type**		**Properties**
userid	varchar	(20)	Primary Key
upassword	varchar	(15)	Not Null
mpassword	varchar	(15)	Not Null

2. **user:** This table contains the information about the user that he enters at the time of the registration. It contains the following fields:

Field Name	**Data type**	**Properties**
userid	varchar(20)	Primary Key
fname	varchar(20)	Not Null
lname	varchar(20)	Not Null
sex	char(1)	Not Null
mstatus	char(1)	Not Null
country	varchar(20)	Not Null

state	varchar(20)	Not Null
city	varchar(20)	Not Null
street	varchar (20)	Not Null
pin	varchar(8)	Not Null
education	varchar(30)	Not Null

3. **pwdrecall:** This table contains the information that is needed when the user forgets his password and uses the password recall option. It contains the following fields:

Field Name	**Data type**	**Properties**
userid	varchar(20)	Primary Key
question	varchar(30)	Not Null
answer	varchar(30)	Not Null
dob	varchar(10)	Not Null

4. **addressbook:** This table contains the address book contact information of all the users. It contains the following fields:

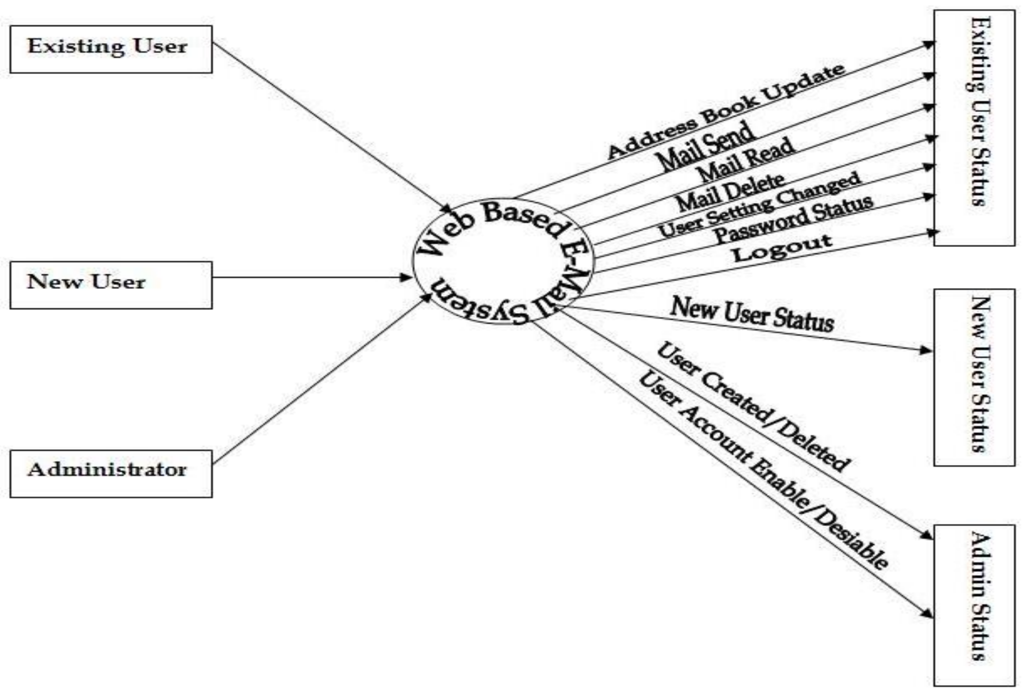
Field Name	**Data type**	**Properties**
userid	varchar(20)	Primary Key
nic_name	varchar(15)	Not Null
email	varchar(40)	Not Null
fname	varchar(20)	May be Null
lname	varchar(20)	May be Null
country	varchar(20)	May be Null

state	varchar(20)	May be Null
city	varchar(20)	May be Null
pin	varchar(8)	May be Null
street	varchar(30)	May be Null
mobile	varchar(20)	May be Null
resi	varchar(20)	May be Null
office	varchar(20)	May be Null

8.5 DATA FLOW DIAGRAMS

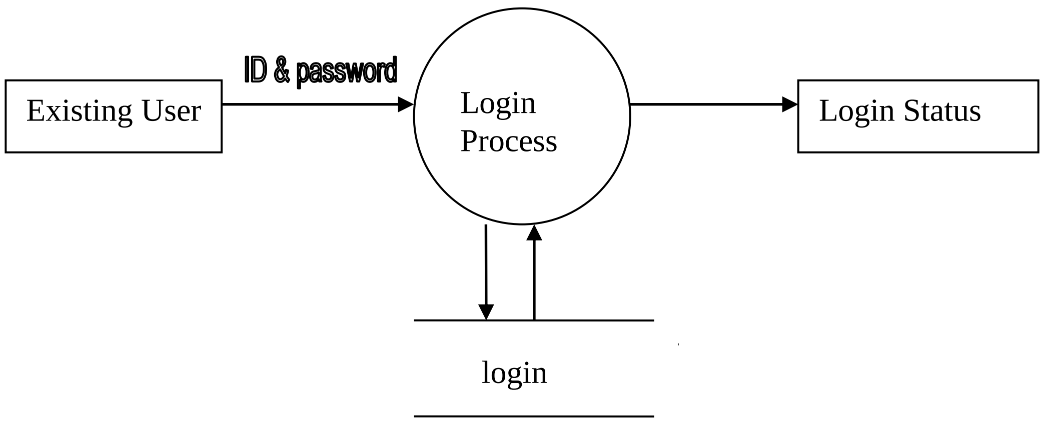
A data flow diagram is a graphical representation that depicts information flow and the transformations that are applied as data move from input to output. The DFD can be partitioned into levels that represent increasing information flow and functional detail. DFD of the E-mail system till the second level are given in the following pages:

8.5.1 CONTEXT LEVEL DFD

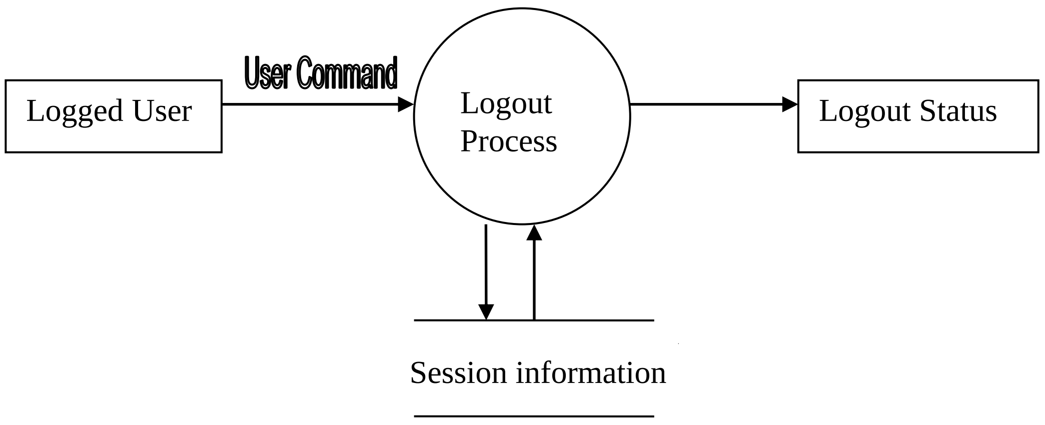


8.5.2 LEVEL ONE DFD

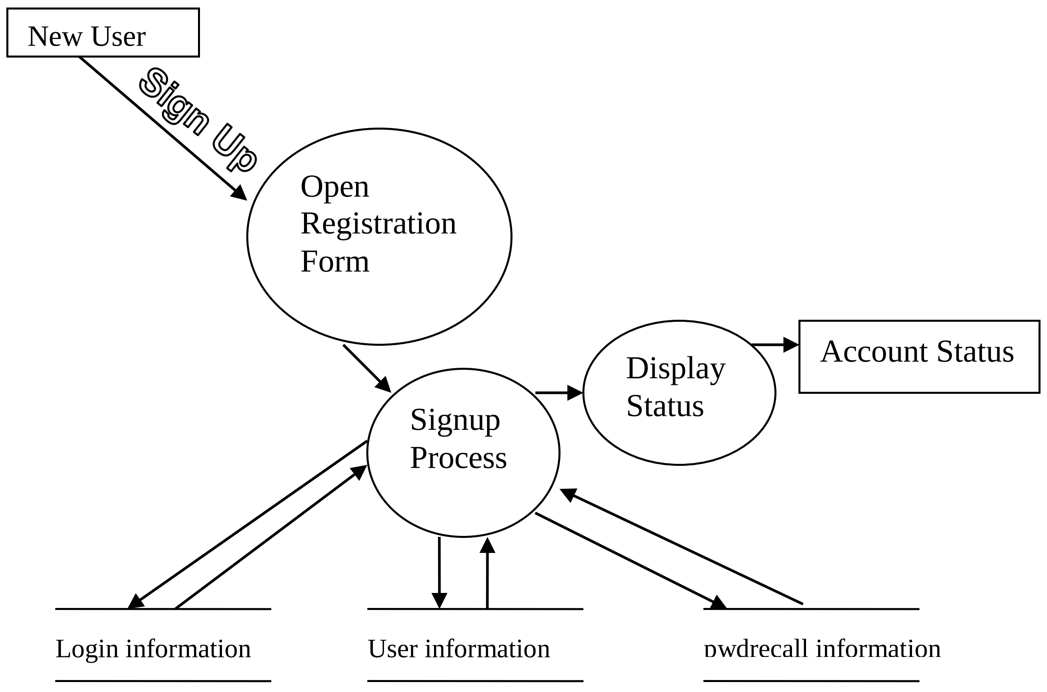
LOGIN PROCESS



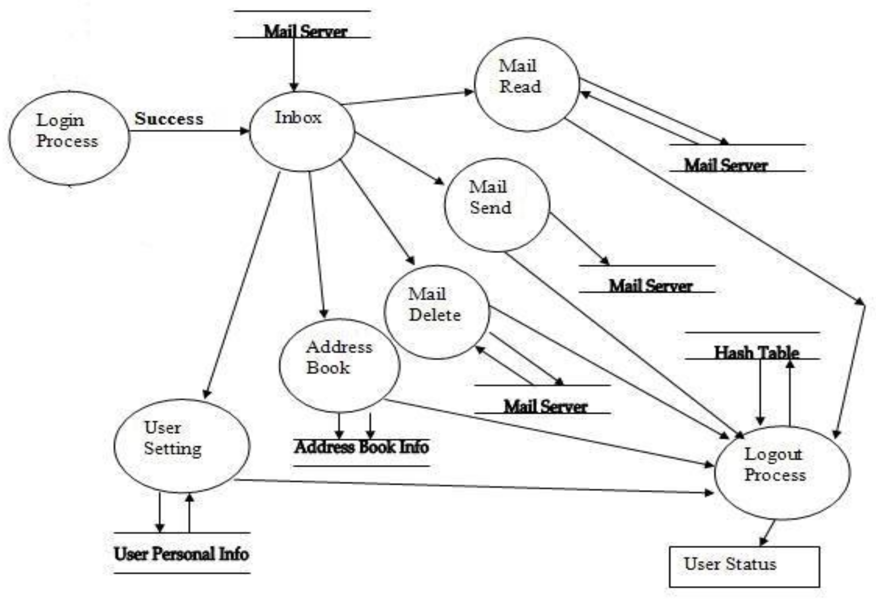
LOGOUT PROCESS



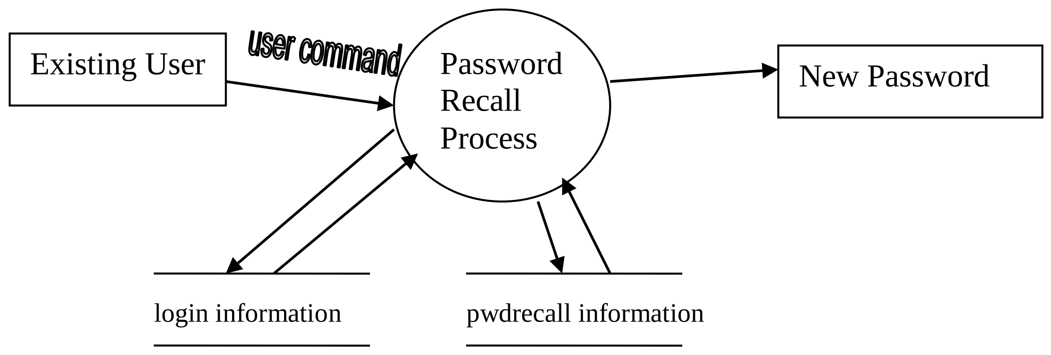
NEW USER REGISTRATION PROCESS



INBOX PROCESSES

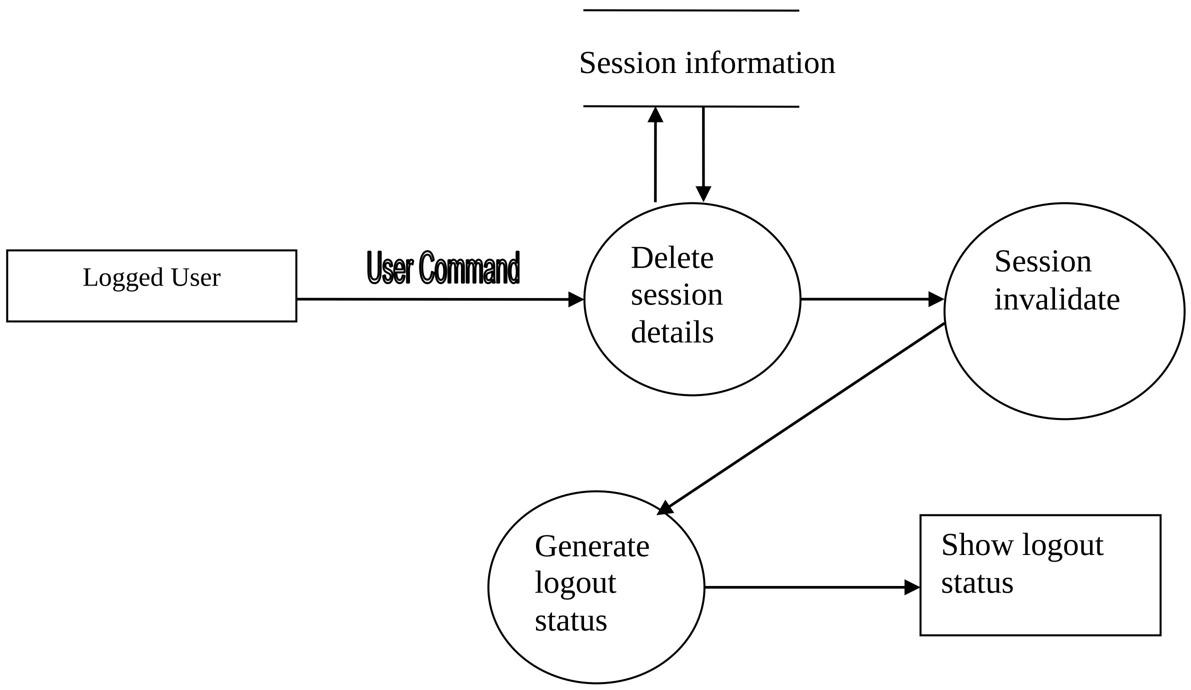


PASSWORD RECALL PROCESS

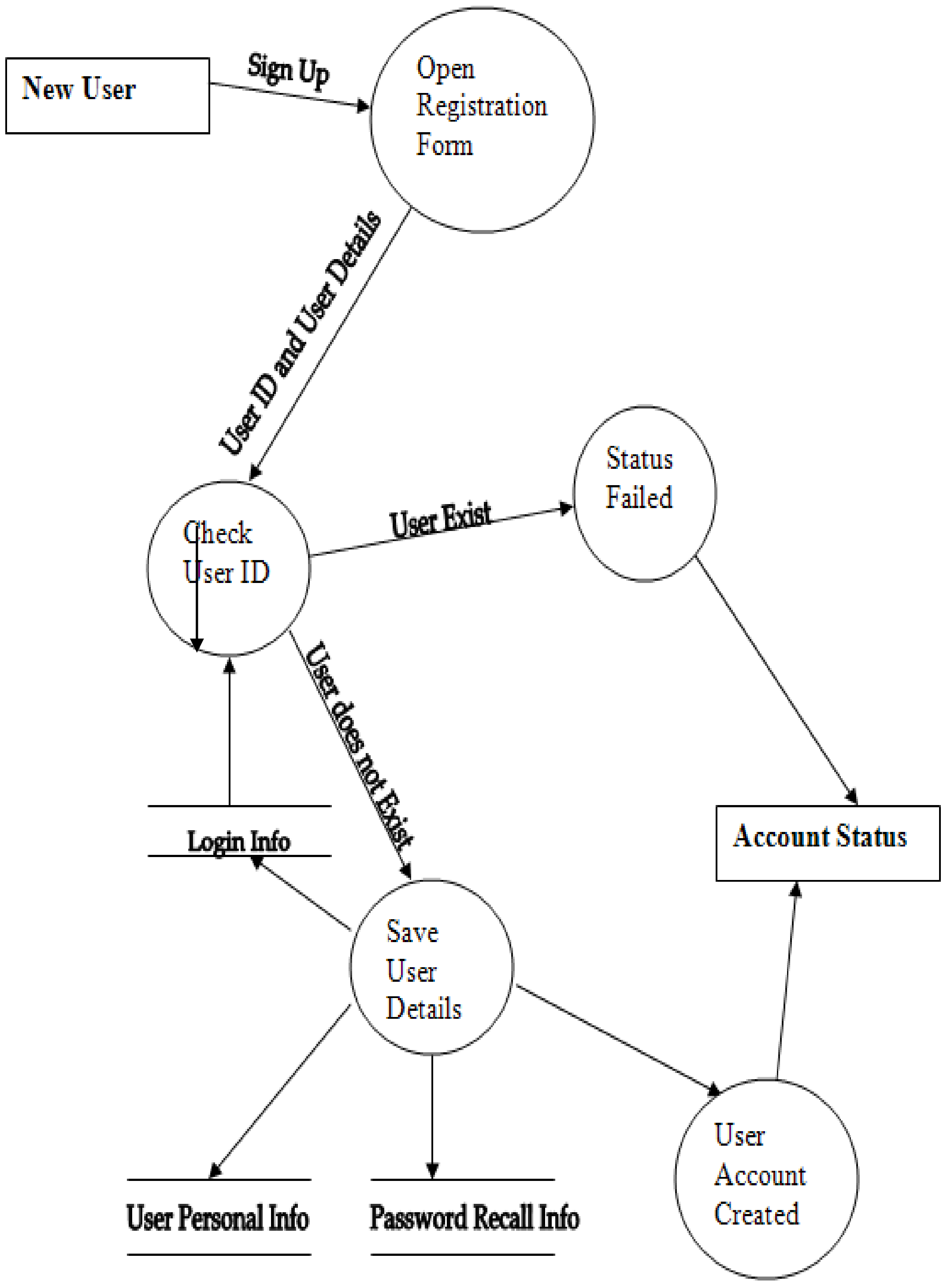


8.5.3 LEVEL TWO DFD

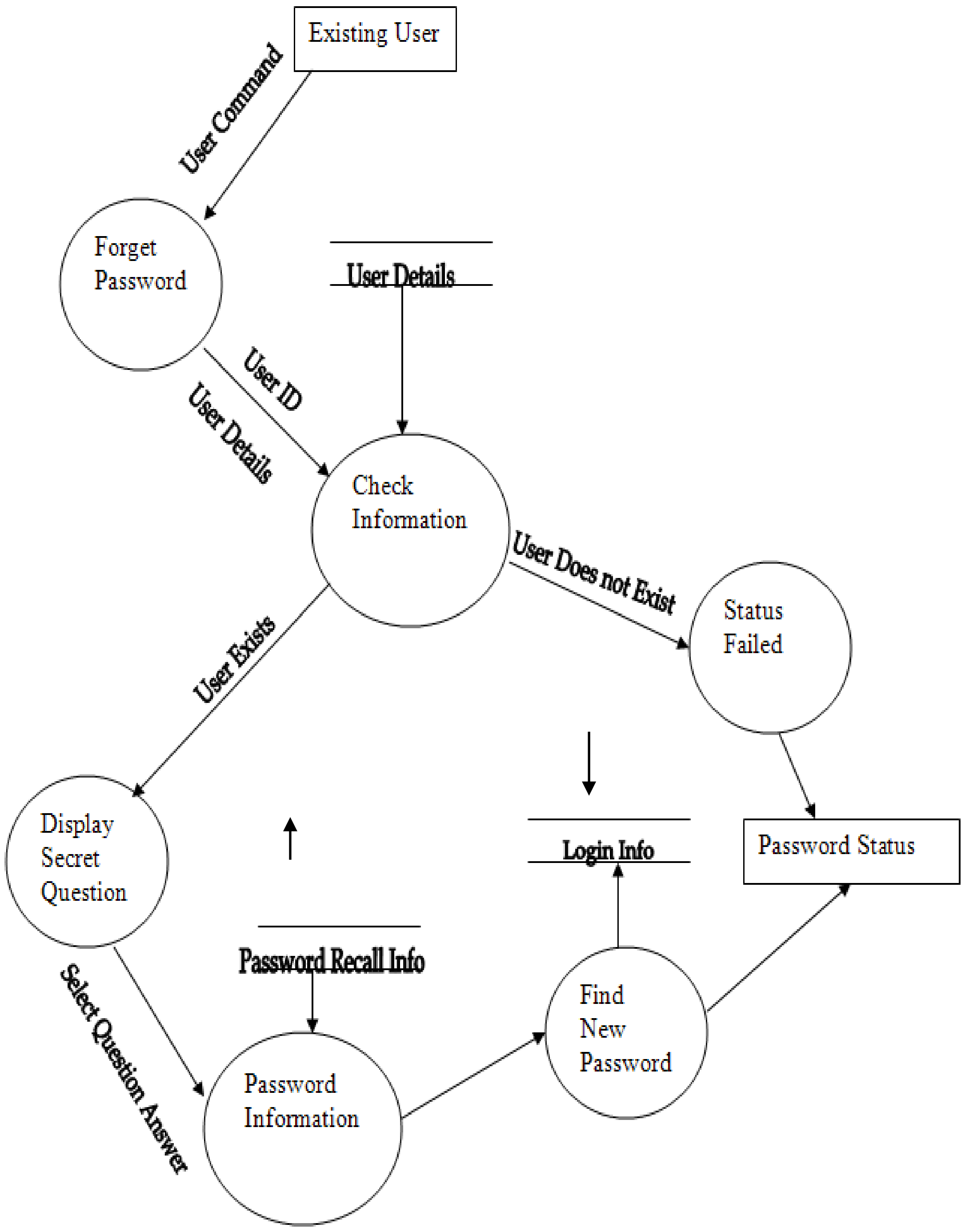
LOGIN PROCESS



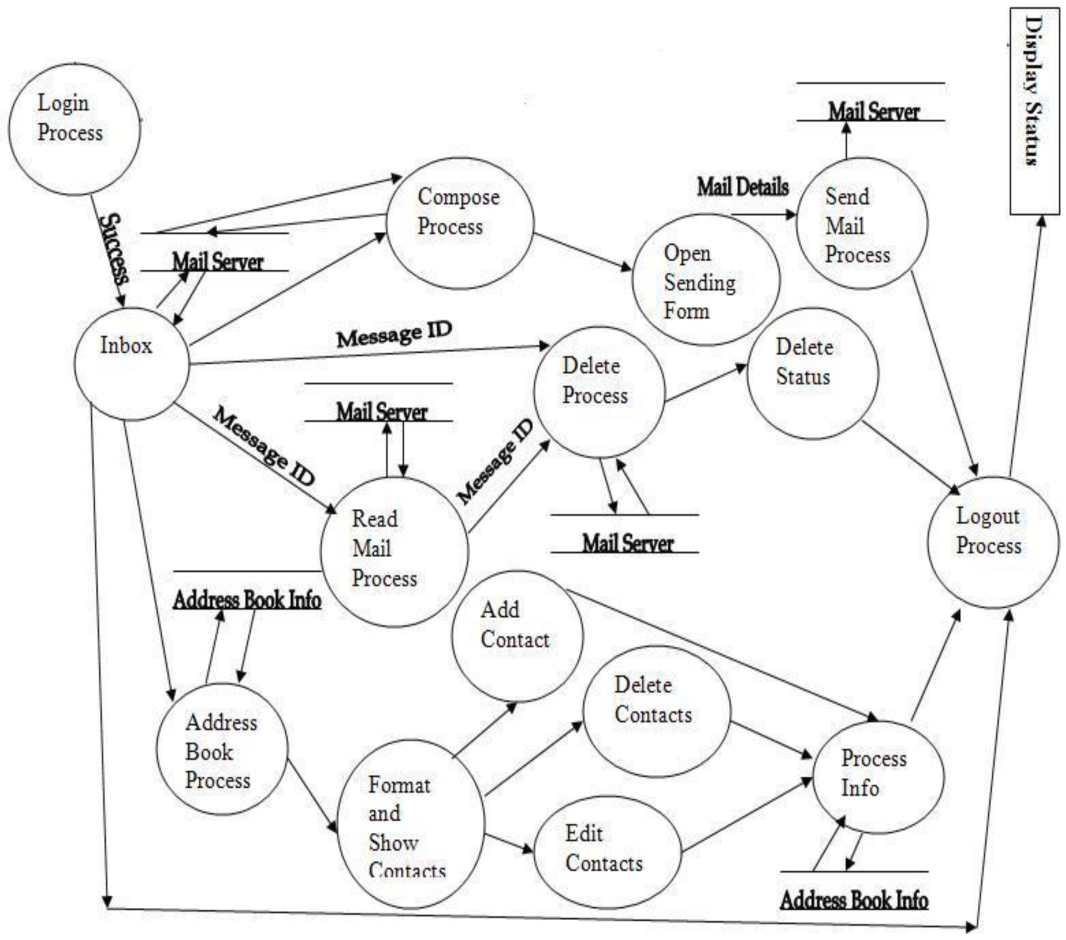
NEW USER REGISTRATION PROCESS



PASSWORD RECALL PROCESS



INBOX PROCESSES



8.6 NUMBER OF MODULES AND DESCRIPTION:

Web based E-mail system contains ten main modules and some sub-modules which along with their short description are as follows:

1. **Registration –** This module is used for registering a new user. This module displays a page which allows a user to check whether a userid exists or is available. If it is available it allows the user to register for that userid. For registering the user has to fill a form. The script then checks whether the form is correctly filled according to the criteria and after that sends it to the server, which process and registers the user.

2. **Login –** This module allows a user to login to the E-mail system. For logging in the system user enters his userid and his password. The values are checked with the database and if it is correct logs the user in and maintains a session for the user in a hash table.

3. **Password Remind –** This module is used for resetting the password of the user who has forgotten his password. This module asks for some private information about the user and also a secret question which it checks with the database after which it changes the user password by a random password which it tells to the user.

4. **Inbox –** This a big module after the user has logged in his mail server password is taken from the database and then userid and password is used to log in the mail system and number of mails, sender name, subject, date etc are read and displayed to the user in a compact form.

5. **Read Mail –** After the user sees the mails he can click on a mail, the program then contacts the mail server and gets all the information about that mail and displays it to the user. The user has the option to go back to the inbox or delete this mail or reply.

6. **Delete Mail –** This module allows the user to delete a mail. To delete the program contacts to the mail server to delete the mail messages. There are two ways to delete messages which are as follows:

   i. **Single Mail** – The user has the option to delete a single mail while reading a mail.

   ii. **Multiple Mails** – The user has the option to select checkboxes beside the headers of the emails in the inbox and then press delete, which deletes all the mails with checked checkboxes.

7. **Compose and Send Mail** – This module is a big module it allows the user to compose a new mail and then sends it to all the recipients and after that displays all the information about the mail send. It also stores the address of all the persons whom mail is send in the address book automatically without any repetition. There are also other features and a lot of error checking in this module.

8. **Change User Setting –** This module allows the user to change his password and other user details that he had supplied at the time of registering with the system. This module is considered to be in high

security zone so the user is asked to enter his password again before he can change his setting.

9. **Address Book –** This module is a big module and it allows the user to maintain his address book. First it displays all the contacts in a compact form which the user can select to change. Following options are available in the address book:

   i. **Add New Contact –** The use has the option to add a new contact by entering certain values in the address book.

   ii. **Delete Contact –** The user has the option to delete an existing contact from the address book.

   iii. **Edit Existing Contact –** The user has the option to change some information or add some more information about some contact.

10. **Logout –** This is an important module the user can logout of the system at any time by pressing the logout option. Logging out deletes the user entry from the server hash table which maintains the user session. The user has to login again in the system before he can do anything if he has pressed logout.

8.7 PROCEDURAL DESIGN OF EACH MODULE:

Few modules like Inbox, Compose and Send Mail etc are very complex and lengthy so it is not possible to describe process logic fully due to limited space. Therefore brief process logic of each module along with their sub-modules of the web based E-mail system is given below:

1. **Registration:**
    1. User selects the option to register.
    2. The controller sends the registration JSP to the user.
    3. User enters a ‘userid’ and then presses ‘Availability’ button.
    4. The page passes the userid to the controller which passes it to the bean to check it with database and tells back to the controller which displays the result with help of JSP.
    5. If the userid is available the user fills the form and submits it.
    6. JavaScript on the page checks if the form is properly filled if properly filled passes it to the controller else shows which fields are not filled properly.
    7. The controller takes the data passes it to the bean that creates the account and tells the controller, the controller checks the result and selects the appropriate JSP and sends it to the user computer displaying that the user was registered or not.

2. **Login:**
    1. User selects the option to login.
    2. The controller sends the login page to the user.
    3. User enters ‘userid’ and ‘password’.

4. The page sends it to the controller which sends it to the bean that checks it with the database and sends back the user mail server password to the controller if login is successful.
5. The controller takes the mail server password from the bean and makes a session entry for the user in the hash table.
6. If the login was successful then the controller shows the JSP of the inbox to the user else shows the login failed JSP to the user.

3. Password Remind:

1. User selects the option for forget password.
2. The controller sends the forget password JSP to the user.
3. User enters userid and tells the date of birth.
4. The user then answers the secret question.
5. The page forwards the information to the controller.
6. Controller passes all the information to the bean which checks it with the database and if correct takes a random password and changes the user password with it and sends it to the controller which takes a JSP and displays the new password to the user.
7. If the information is incorrect the controller takes another JSP to display the appropriate error.

4. Inbox:

1. The controller takes the userid and mail password and passes it to the mail server requesting about the number of mails and their information.
2. The mail server sends all the mail information which the controller passes to the inbox JSP.
3. Inbox JSP takes all the information and formats it in a compact form and displays it to the user.

5. Read Mail:

1. The user clicks on a mail.
2. The page passes its message id to the controller.
3. The controller takes the userid and mail server password from the hash table and passes it along with the message id requesting the mail content.
4. The mail server replies back to the controller and pass it the mail data.
5. The controller takes the data and passes it to the read mail JSP.
6. Read mail JSP formats the data and displays it to the user.

6. Delete Mail:

Deleting can be done in two ways:

i. **Single Mail:**

1. User presses the delete button while reading an E-mail.
2. The page passes the message id to the controller.
3. The controller takes the userid and mail server password from the hash table and passes it to the mail server along with the delete requesting mail flag enabled true.
4. The mail server deletes the mail.
5. The controller refreshes the content of the inbox from the mail server using the inbox module and shows it to the user.

ii. Multiple Mails:

1. User checks the checkboxes of the mails he wants to delete in the inbox and selects delete.
2. The page passes all the message ids to the controller.
3. The controller takes the userid and mail server password from the hash table and passes it to the mail server along with the delete requesting mail flag enabled true for all the mails to be deleted.
4. The mail server deletes the mail.
5. The controller refreshes the content of the inbox from the mail server using the inbox module and shows it to the user.

7. Compose and Send Mail:

1. The user clicks on the option to compose mail.
2. The controller sends the request for the users address book addresses to the database and receives the addresses.
3. Controller passes the addresses to the compose JSP.
4. Compose JSP formats the addresses and shows the compose page with the addresses.
5. The user has the option to click on addresses and they get entered in the address field of the form.
6. The user enters the subject, message and other details and clicks on send.
7. The page sends all information to the controller, which sends it to the mail server with the ‘userid’ and mail server password.
8. Controller also checks the addresses with the address book of the user and adds all the new addresses to the address book.

9. The controller displays the page with the appropriate information to who was this mail send and what addressees are added to the address book.

8. Change User Setting:

1. The user clicks on the option to change his setting.
2. The controller handles the request and sends a page requesting the user password.
3. The user enters the password and clicks submit.
4. The page sends the password to the controller, which sends it to the bean to check it with the database, which sends the result back to the controller.
5. If the password is correct the controller sends the setting change page to the user.
6. The user can change his password as well as all the information about him and clicks save.
7. The JavaScript checks the data and if filled correctly sends it to the controller.
8. The controller sends it to the bean which updates the database and sends the result of the update to the controller.
9. The controller takes the appropriate page according to the result and displays it to the user.

9. Address Book:

1. The user clicks on the option to go to the address book.
2. The page passes the request to the controller which sends the request for the address information of the address book to the database.
3. The database sends the data back.
4. The controller passes the data to the address book JSP, which formats it and displays all the addresses in a compact form to the user.
5. The user then selects any of the three option then displayed which do the following tasks:

    i. **Add New Contact:**

    1. The user clicks on the 'add new contact' option.
    2. The page sends the request to the controller which sends the 'add new contact' JSP to the user.
    3. The user enters all the details and clicks on save.
    4. The page passes the data to the controller which passes it to the bean.
    5. Bean saves the contact in the database and sends the result to the controller.
    6. Controller then refreshes the addresses with the database and sends the address book JSP to the user.

    ii. **<u>Delete Contact:</u>**

    1. User checks one or more checkboxes displayed in the address book page and presses delete.

2. Page sends the request to the controller with the id of the addresses to be deleted.
3. The controller sends the request to the bean which updates the database and sends the result to the controller.
4. Controller then refreshes the addresses with the database and sends the address book JSP to the user.

iii. Edit Existing Contact:

1. User selects a contact and presses 'edit contact'.
2. The page sends request to the controller, which requests more information about that contact from the database
3. The controller then passes that information to the 'edit contact' JSP which formats and displays it.
4. The user then adds some information deletes or updates it and clicks save.
5. The page sends data to the controller, which sends it to the database. The database saves it in the database and sends the result to the controller.
6. Controller then refreshes the addresses with the database and sends the address book JSP to the user.

10. Logout:

1. The user clicks on the logout link from any page.
2. The Page passes the request to the controller.
3. The controller deletes the session entry of the user from the hash table and the session of the user ends.

4. The controller then shows the login JSP to the user stating the user has logged out.

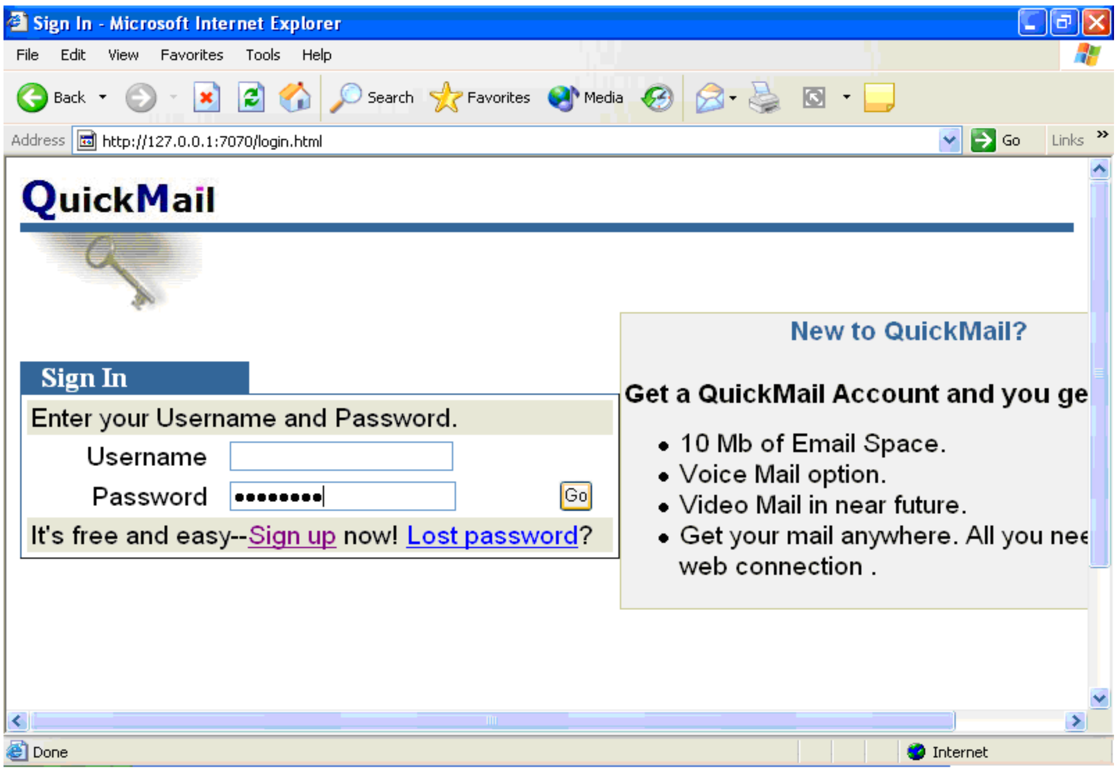
USER INTERFACE DESIGN

9 USER INTERFACE DESIGN

User interface has to be very easy so that e-mail website can be used by persons who are not very comfortable with computers so a great care was taken to make the web interface as simple as possible. It was designed using the layout as most other e-mail services provide, like Rediffmail and Yahoomail so it will be very easy to use for anyone who has used e-mail before.

Screenshots of the various WebPages have been given in the following pages:

9.1 LOGIN PAGE


Sign In - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/login.html
Go Links
Done
Internet

QuickMail

Sign In

Enter your Username and Password.

Username

Password ●●●●●●●●● Go

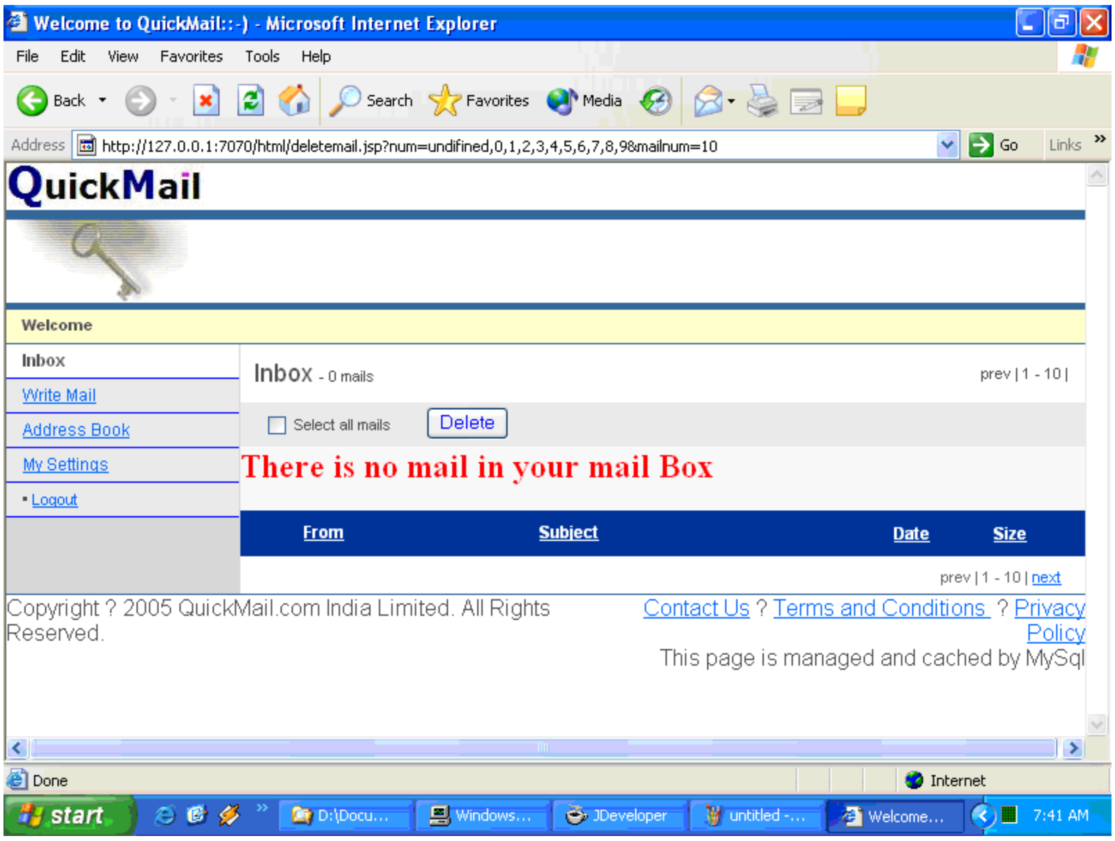
It's free and easy--Sign up now! Lost password?

New to QuickMail?

Get a QuickMail Account and you ge

- 10 Mb of Email Space.
- Voice Mail option.
- Video Mail in near future.
- Get your mail anywhere. All you nee web connection .

9.2 INBOX WITH NO MAIL



Welcome to QuickMail::-) - Microsoft Internet Explorer

Address http://127.0.0.1:7070/html/deletemail.jsp?num=undifined,0,1,2,3,4,5,6,7,8,9&mailnum=10

QuickMail

Welcome

- Inbox
- Write Mail
- Address Book
- My Settings
- Logout

Inbox - 0 mails   prev | 1 - 10 |

Select all mails Delete

There is no mail in your mail Box

From	Subject	Date	Size

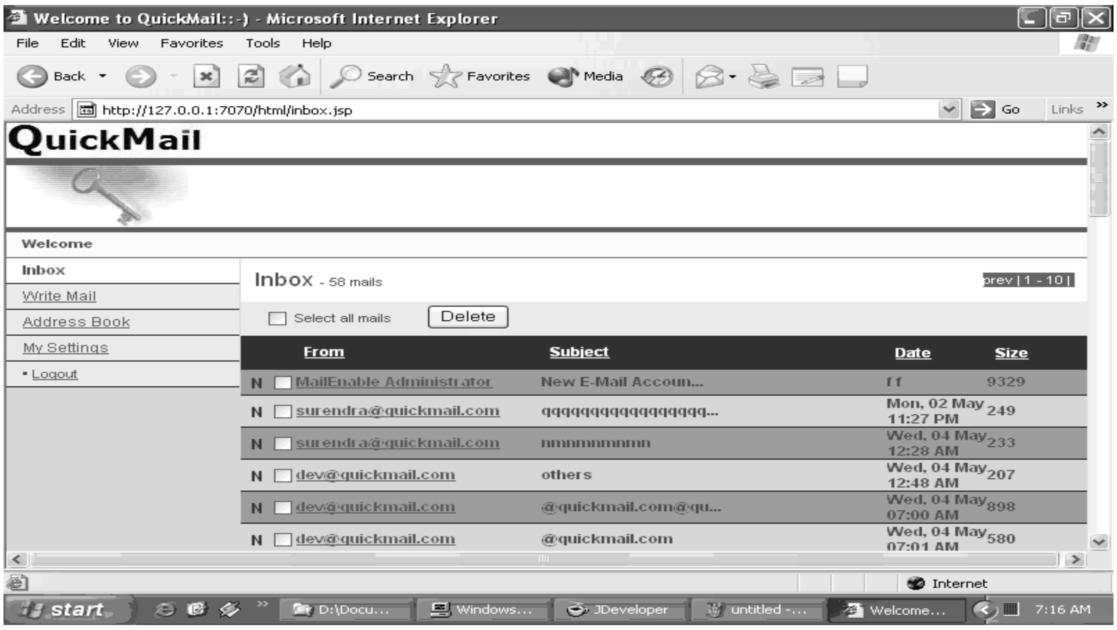
prev | 1 - 10 | next

Copyright ? 2005 QuickMail.com India Limited. All Rights Reserved.

Contact Us ? Terms and Conditions ? Privacy Policy

This page is managed and cached by MySql

9.3 INBOX WITH MAIL



Welcome to QuickMail::-) - Microsoft Internet Explorer

Address http://127.0.0.1:7070/html/inbox.jsp

QuickMail

Welcome

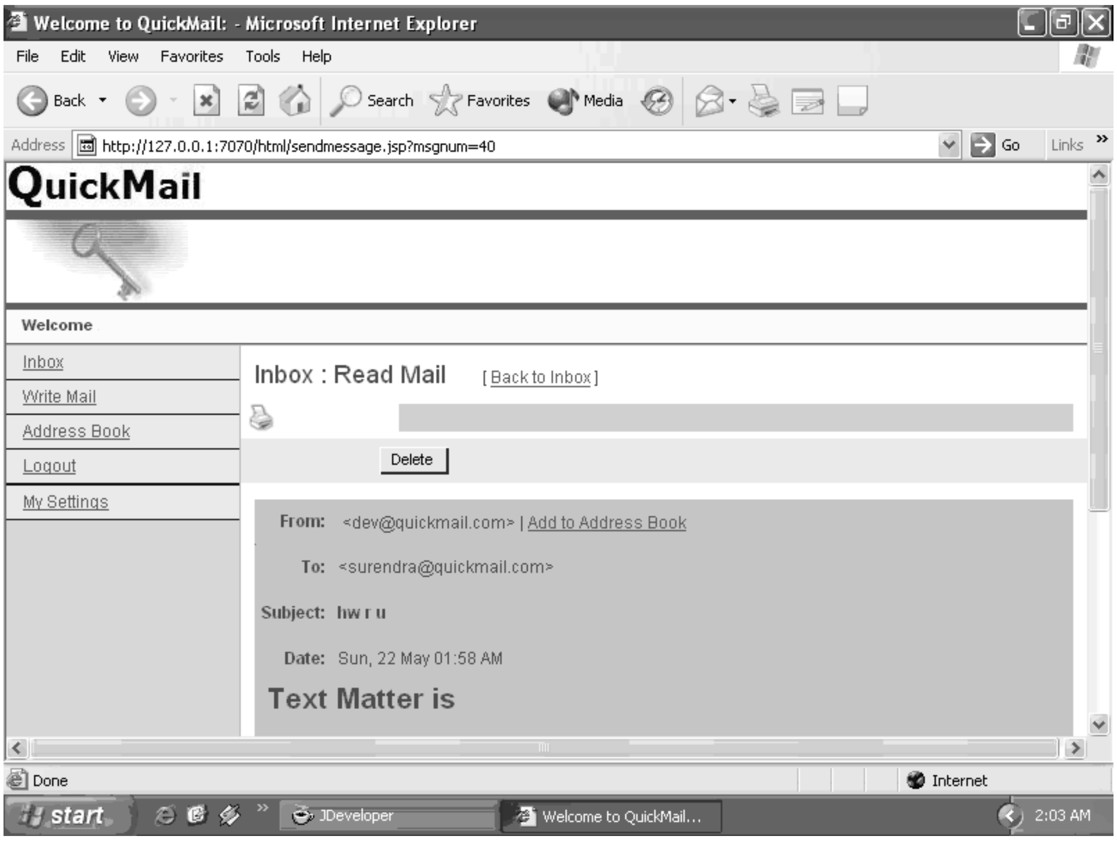
- Inbox
- Write Mail
- Address Book
- My Settings
- Logout

Inbox - 58 mails   prev | 1 - 10 |

Select all mails Delete

	From	Subject	Date	Size
N	MailEnable Administrator	New E-Mail Accoun...	f f	9329
N	surendra@quickmail.com	qqqqqqqqqqqqqqqq...	Mon, 02 May 11:27 PM	249
N	surendra@quickmail.com	nmnmnmnmn	Wed, 04 May 12:28 AM	233
N	dev@quickmail.com	others	Wed, 04 May 12:48 AM	207
N	dev@quickmail.com	@quickmail.com@qu...	Wed, 04 May 07:00 AM	898
N	dev@quickmail.com	@quickmail.com	Wed, 04 May 07:01 AM	580

9.4 READ MAIL


Welcome to QuickMail: - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/html/sendmessage.jsp?msgnum=40
Go Links
QuickMail
Welcome
Inbox
Write Mail
Address Book
Logout
My Settings
Delete
Done
Internet
start
JDeveloper
Welcome to QuickMail...
2:03 AM

Inbox : Read Mail [Back to Inbox]

From: <dev@quickmail.com> | Add to Address Book

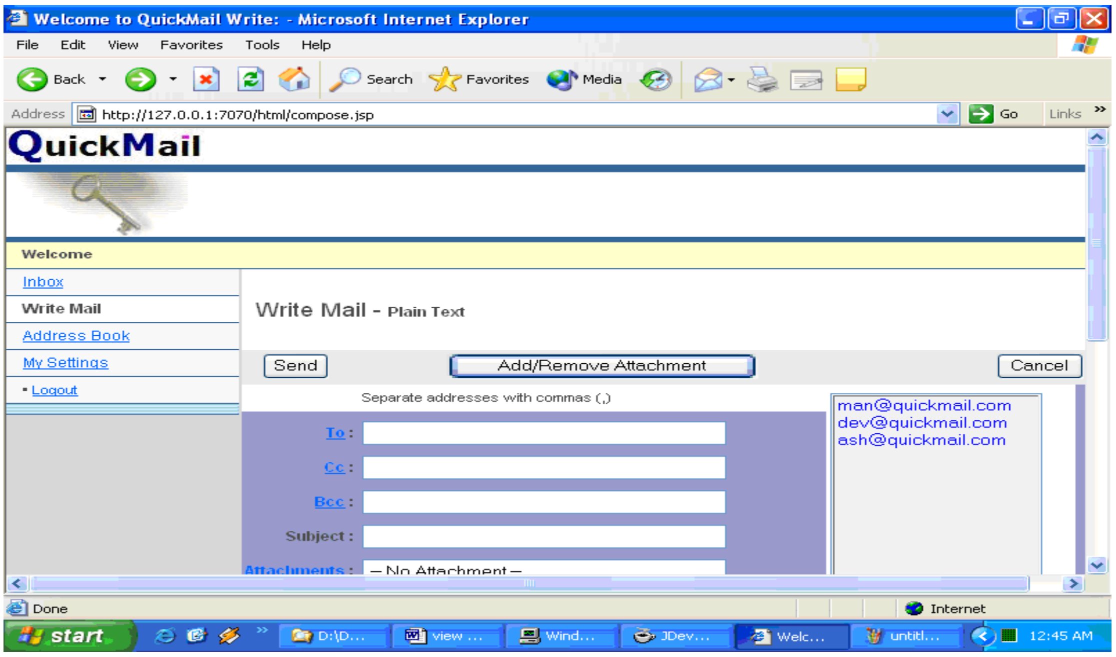
To: <surendra@quickmail.com>

Subject: hw r u

Date: Sun, 22 May 01:58 AM

Text Matter is

9.5 COMPOSE PAGE



Write Mail - Plain Text

Separate addresses with commas (,)

To :

Cc :

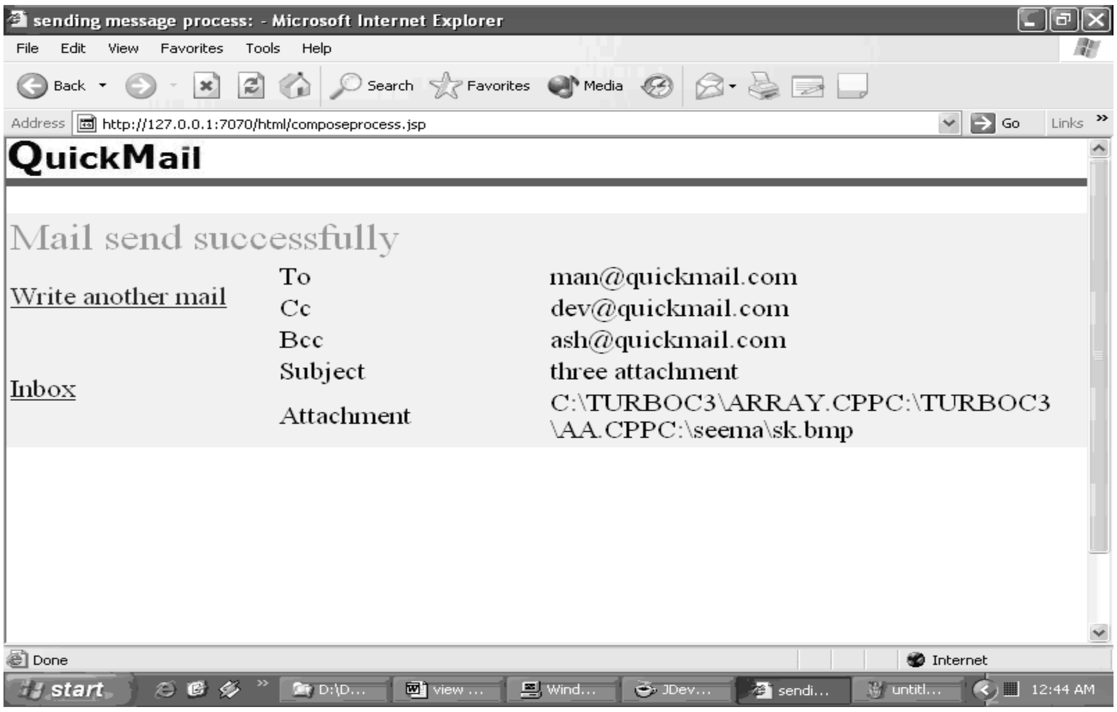
Bcc :

Subject :

Attachments : -- No Attachment --

man@quickmail.com
dev@quickmail.com
ash@quickmail.com

9.6 MAIL SEND SUCCESSFULLY

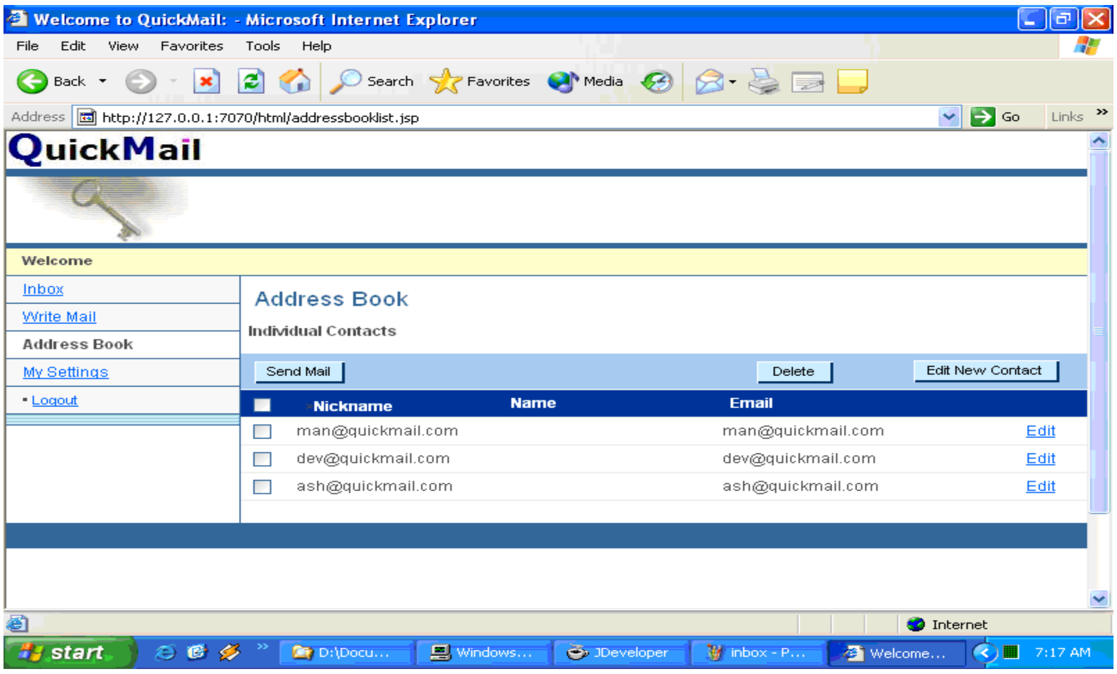

sending message process: - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/html/composeprocess.jsp Go Links
QuickMail

Mail send successfully

Write another mail	To	man@quickmail.com
	Cc	dev@quickmail.com
	Bcc	ash@quickmail.com
Inbox	Subject	three attachment
	Attachment	C:\TURBOC3\ARRAY.CPPC:\TURBOC3 \AA.CPPC:\seema\sk.bmp

Done
Internet
start D:\D... view ... Wind... JDev... sendi... untitl... 12:44 AM

9.7 CONTACTS LIST


Welcome to QuickMail: - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/html/addressbooklist.jsp Go Links
QuickMail
Welcome
Inbox
Write Mail
Address Book
My Settings
Logout

Address Book

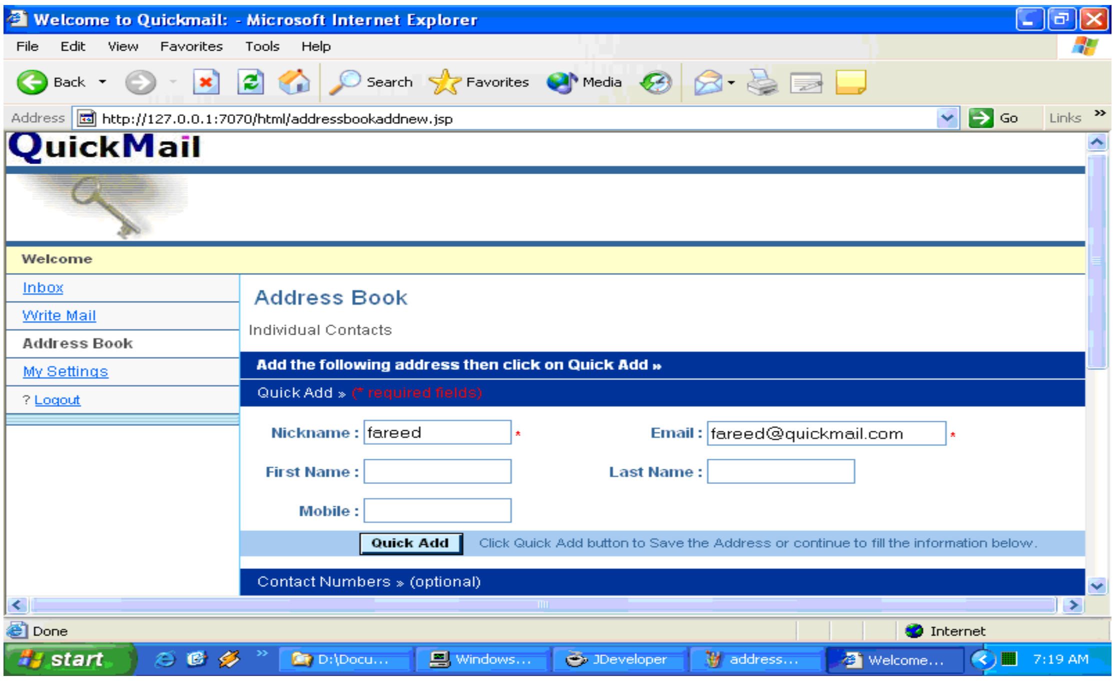
Individual Contacts

Send Mail
Delete
Edit New Contact

	Nickname	Name	Email	
☐	man@quickmail.com		man@quickmail.com	Edit
☐	dev@quickmail.com		dev@quickmail.com	Edit
☐	ash@quickmail.com		ash@quickmail.com	Edit

Internet
start D:\Docu... Windows... JDeveloper inbox - P... Welcome... 7:17 AM

9.8 EDITING CONTACT



Welcome to Quickmail: - Microsoft Internet Explorer

Address: http://127.0.0.1:7070/html/addressbooknew.jsp

QuickMail

Welcome

Inbox
Write Mail
Address Book
My Settings
? Logout

Address Book

Individual Contacts

Add the following address then click on Quick Add »

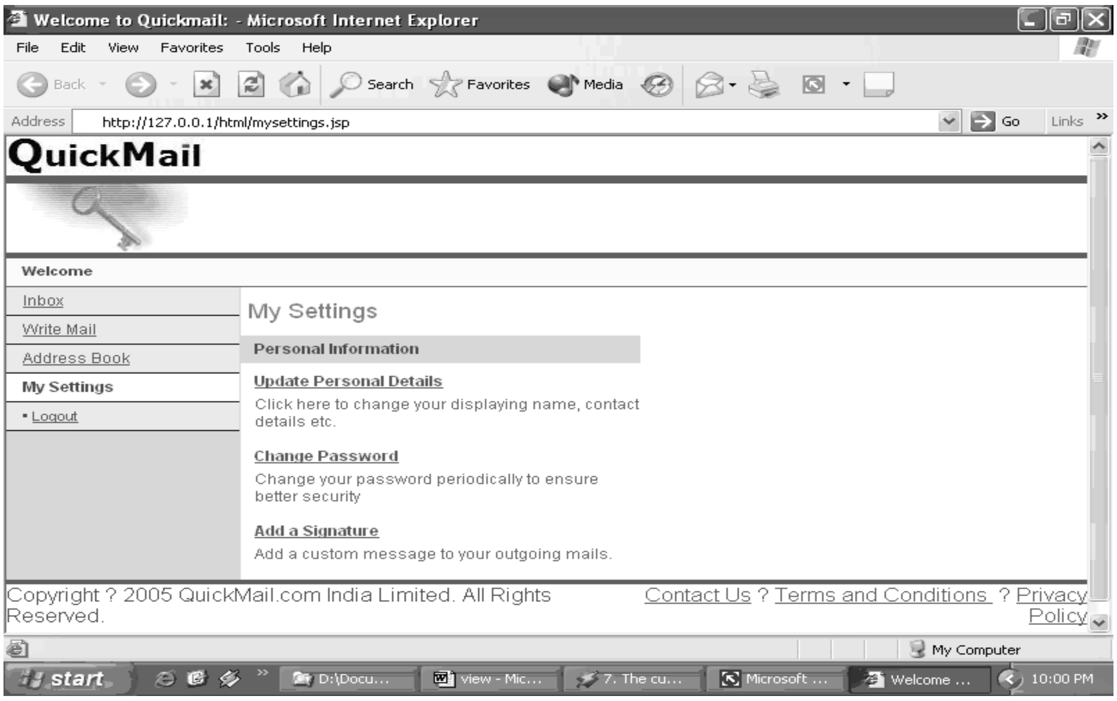
Quick Add » (* required fields)

Nickname : fareed *
Email : fareed@quickmail.com *
First Name :
Last Name :
Mobile :

Quick Add — Click Quick Add button to Save the Address or continue to fill the information below.

Contact Numbers » (optional)

9.9 SETTINGS PAGE



Welcome to Quickmail: - Microsoft Internet Explorer

Address: http://127.0.0.1/html/mysettings.jsp

QuickMail

Welcome

Inbox
Write Mail
Address Book
My Settings
▪ Logout

My Settings

Personal Information

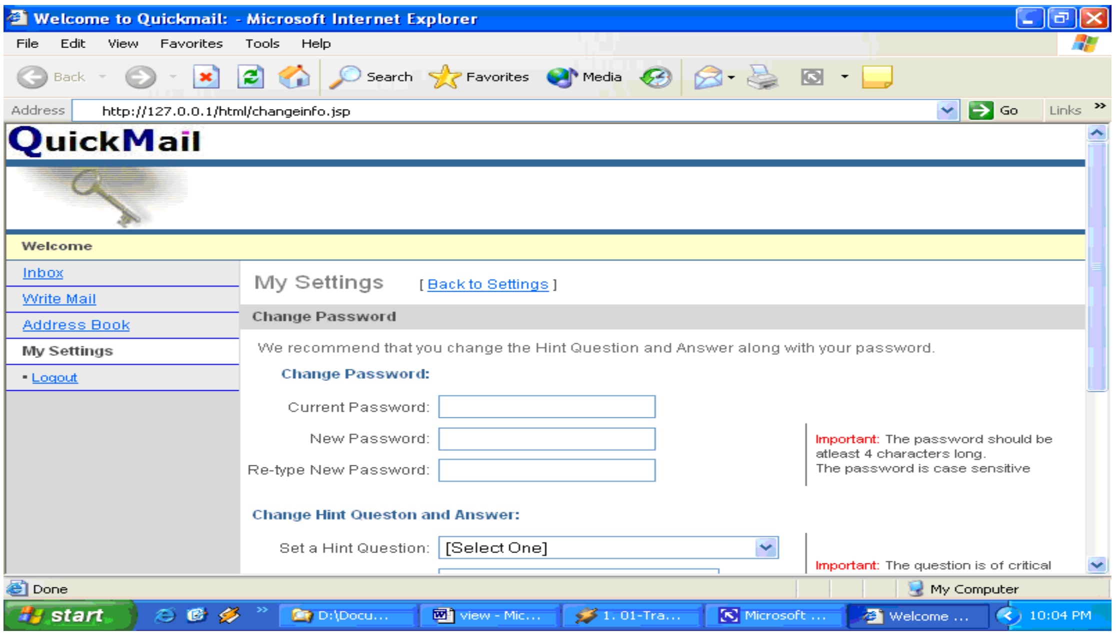
Update Personal Details
Click here to change your displaying name, contact details etc.

Change Password
Change your password periodically to ensure better security

Add a Signature
Add a custom message to your outgoing mails.

Copyright ? 2005 QuickMail.com India Limited. All Rights Reserved.
Contact Us ? Terms and Conditions ? Privacy Policy

9.10 SETTING CHANGE PAGE


Welcome to Quickmail: - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1/html/changeinfo.jsp
Go Links
QuickMail
Welcome
Inbox
Write Mail
Address Book
My Settings
Logout
Done
My Computer
start
10:04 PM

My Settings [Back to Settings]

Change Password

We recommend that you change the Hint Question and Answer along with your password.

Change Password:

Current Password:

New Password:

Re-type New Password:

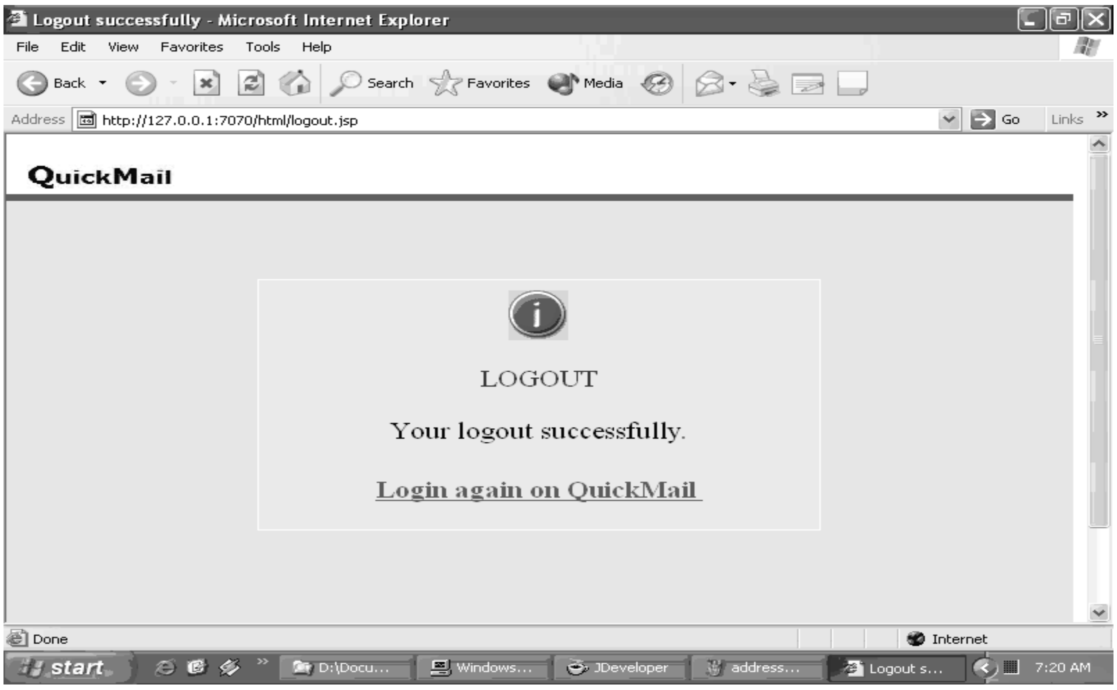
Important: The password should be atleast 4 characters long.
The password is case sensitive

Change Hint Queston and Answer:

Set a Hint Question: [Select One]

Important: The question is of critical

9.11 LOGOUT PAGE

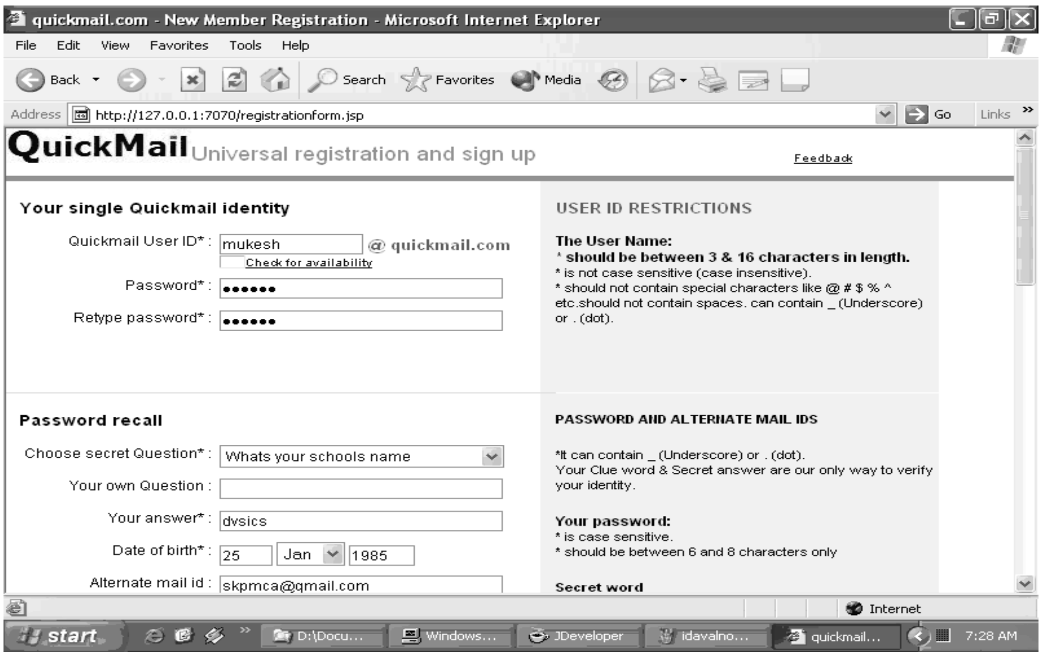

Logout successfully - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/html/logout.jsp
Go Links
QuickMail
Done
Internet
start
7:20 AM

LOGOUT

Your logout successfully.

Login again on QuickMail

9.12 REGISTRATION PAGE


quickmail.com - New Member Registration - Microsoft Internet Explorer

Address http://127.0.0.1:7070/registrationform.jsp

QuickMail Universal registration and sign up

Feedback

Your single Quickmail identity

Quickmail User ID* :	mukesh @ quickmail.com
	Check for availability
Password* :	••••••
Retype password* :	••••••

USER ID RESTRICTIONS

The User Name:
*** should be between 3 & 16 characters in length.**
* is not case sensitive (case insensitive).
* should not contain special characters like @ # $ % ^ etc.should not contain spaces. can contain _ (Underscore) or . (dot).

Password recall

Choose secret Question* :	Whats your schools name
Your own Question :	
Your answer* :	dvsics
Date of birth* :	25 Jan 1985
Alternate mail id :	skpmca@gmail.com

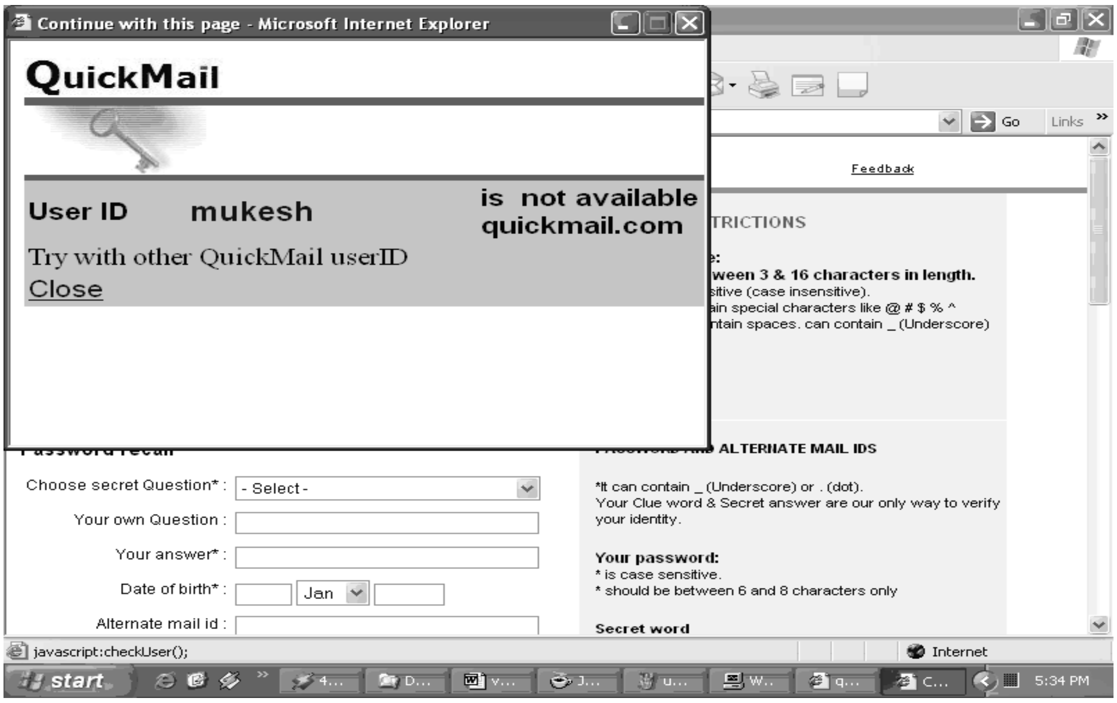
PASSWORD AND ALTERNATE MAIL IDS

*It can contain _ (Underscore) or . (dot).
Your Clue word & Secret answer are our only way to verify your identity.

Your password:
* is case sensitive.
* should be between 6 and 8 characters only

Secret word

9.13 DUPLICATE USERID CHECK PAGE


Continue with this page - Microsoft Internet Explorer

QuickMail

User ID mukesh is not available quickmail.com

Try with other QuickMail userID

Close

Feedback

TRICTIONS

ween 3 & 16 characters in length.
sitive (case insensitive).
ain special characters like @ # $ % ^
ntain spaces. can contain _ (Underscore)

ALTERNATE MAIL IDS

Choose secret Question* :	- Select -
Your own Question :	
Your answer* :	
Date of birth* :	Jan
Alternate mail id :	

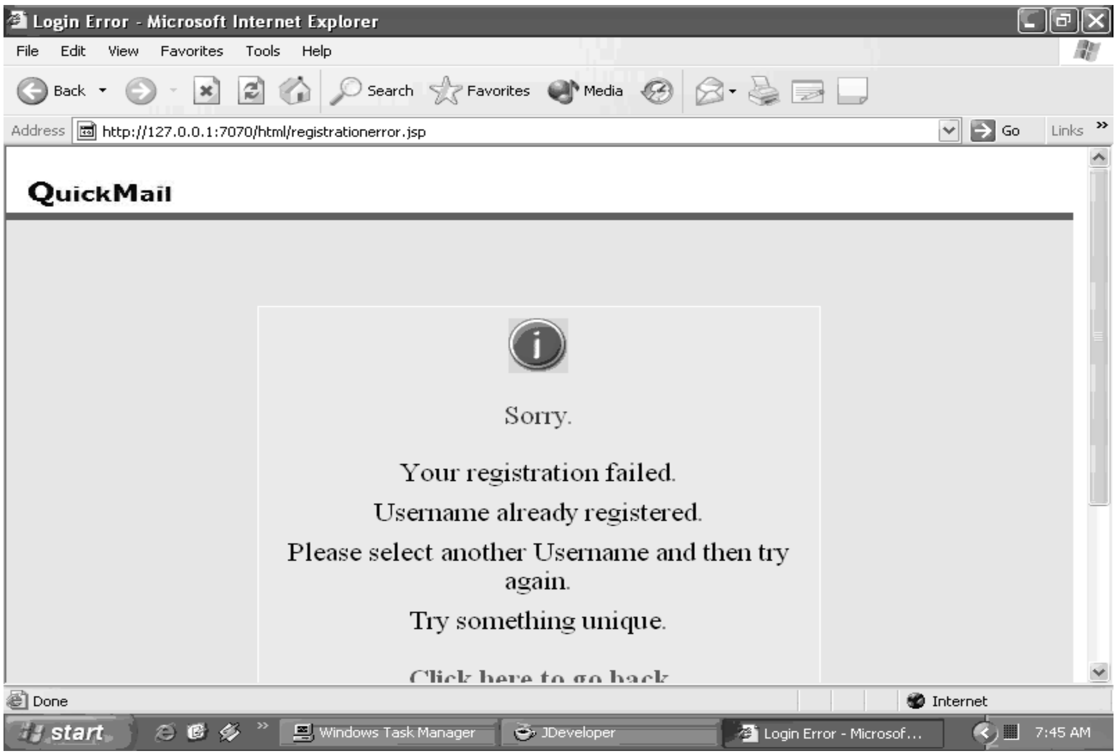
*It can contain _ (Underscore) or . (dot).
Your Clue word & Secret answer are our only way to verify your identity.

Your password:
* is case sensitive.
* should be between 6 and 8 characters only

Secret word

javascript:checkUser();

9.14 ERROR REGISTRATION FAILED


Login Error - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/html/registrationerror.jsp
Go Links
QuickMail

Sorry.

Your registration failed.

Username already registered.

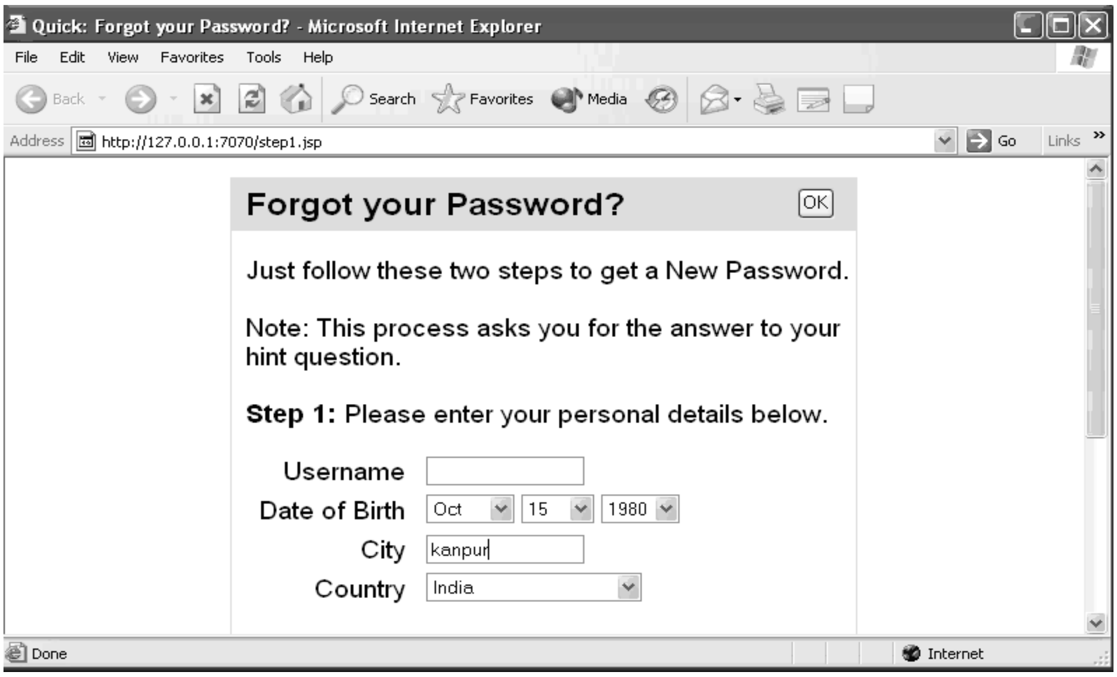
Please select another Username and then try again.

Try something unique.

Click here to go back

Done
Internet
start
Windows Task Manager
JDeveloper
Login Error - Microsof...
7:45 AM

9.15 PASSWORD RESET STEP ONE


Quick: Forgot your Password? - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/step1.jsp
Go Links

Forgot your Password? OK

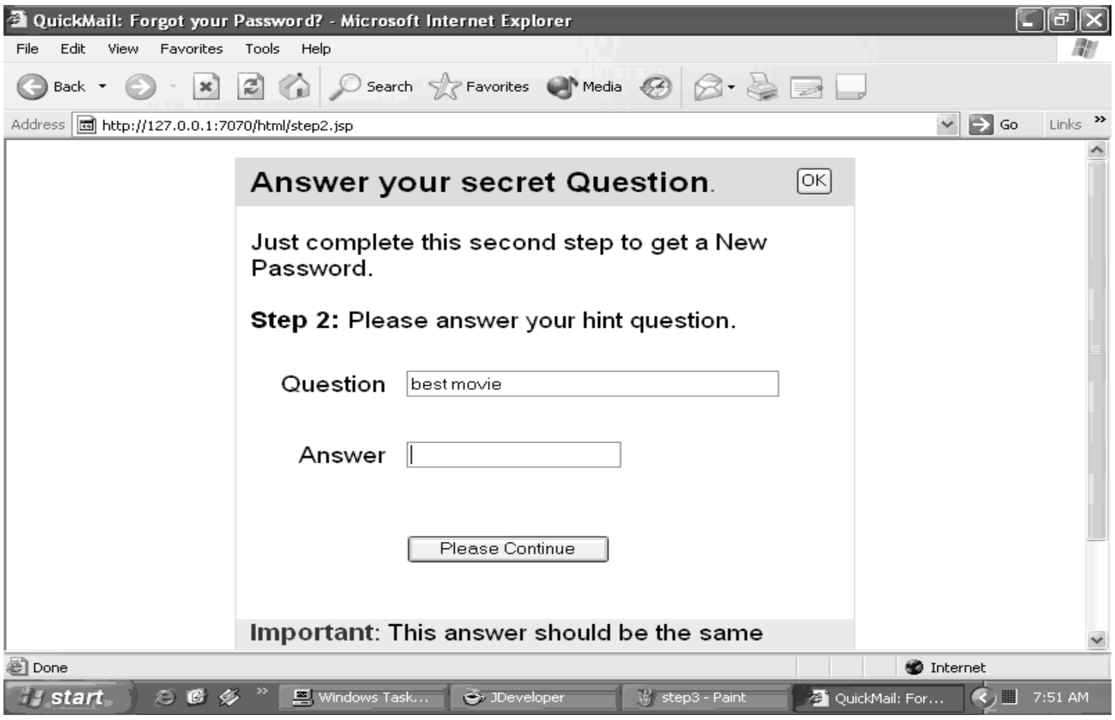
Just follow these two steps to get a New Password.

Note: This process asks you for the answer to your hint question.

Step 1: Please enter your personal details below.

Username	
Date of Birth	Oct 15 1980
City	kanpur
Country	India

9.16 PASSWORD RESET STEP TWO


QuickMail: Forgot your Password? - Microsoft Internet Explorer
http://127.0.0.1:7070/html/step2.jsp

Answer your secret Question.

Just complete this second step to get a New Password.

Step 2: Please answer your hint question.

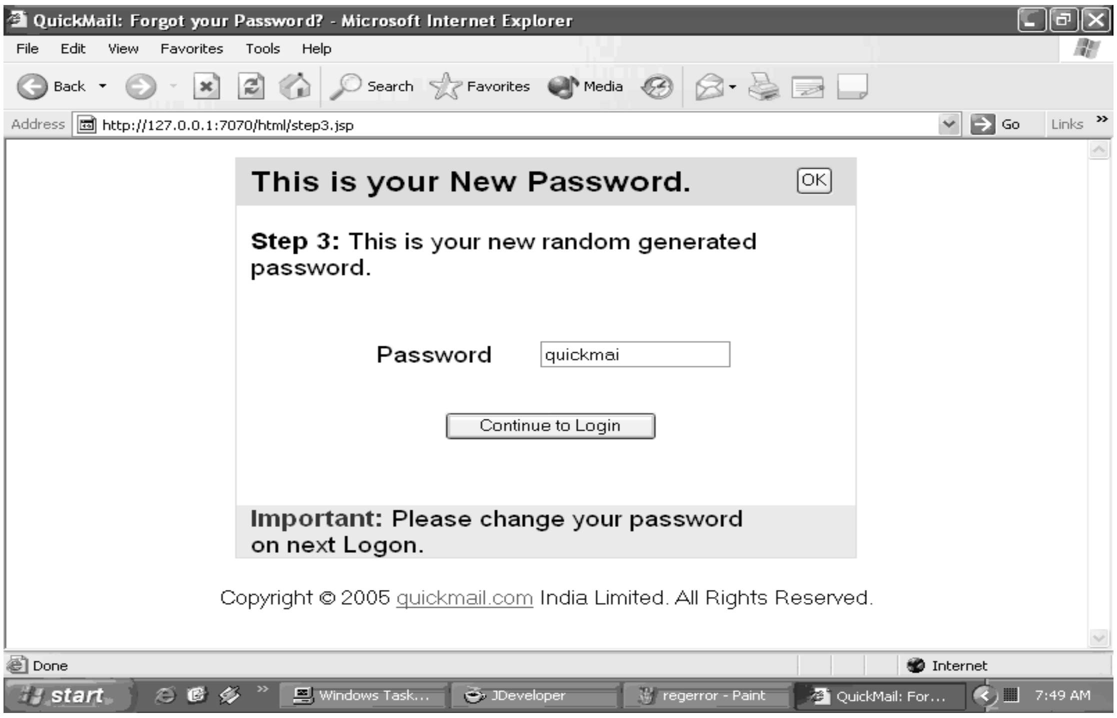
Question: best movie

Answer:

Please Continue

Important: This answer should be the same

9.17 PASSWORD RESET STEP THREE


QuickMail: Forgot your Password? - Microsoft Internet Explorer
http://127.0.0.1:7070/html/step3.jsp

This is your New Password.

Step 3: This is your new random generated password.

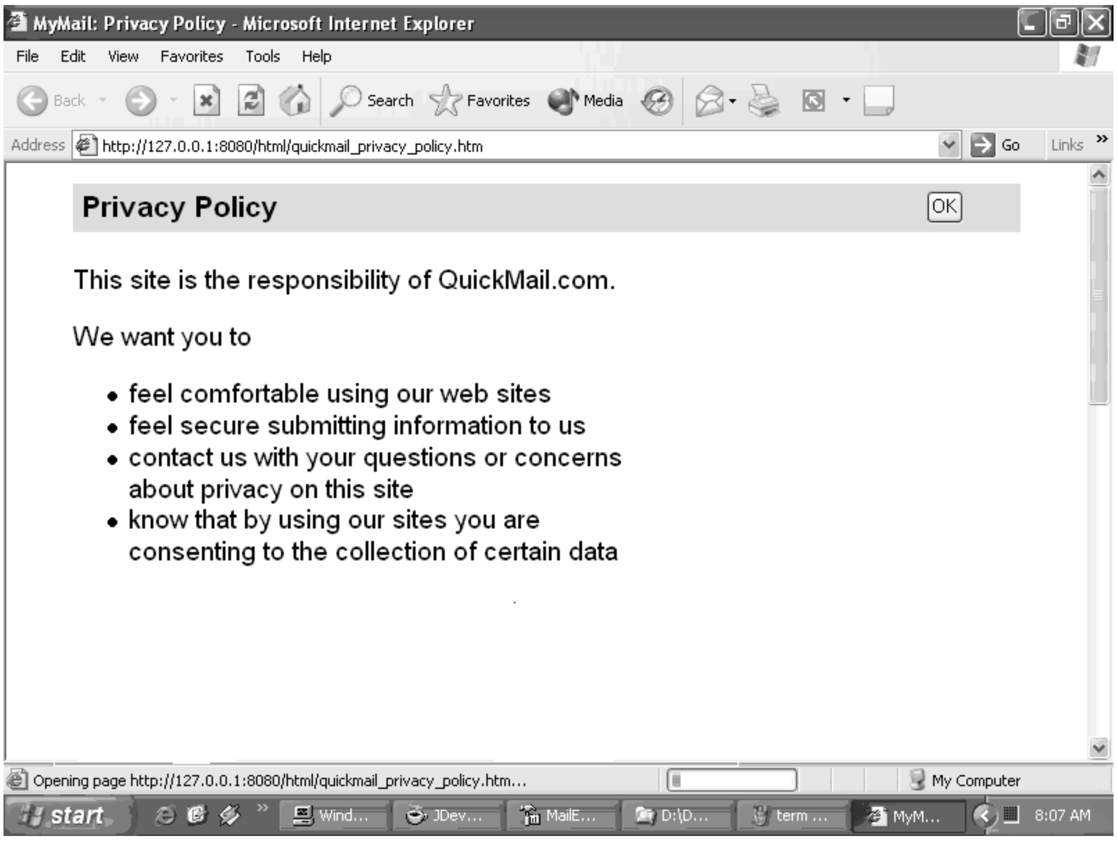
Password: quickmai

Continue to Login

Important: Please change your password on next Logon.

Copyright © 2005 quickmail.com India Limited. All Rights Reserved.

9.18 PRIVACY POLICY PAGE


MyMail: Privacy Policy - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:8080/html/quickmail_privacy_policy.htm
Go Links
OK
Opening page http://127.0.0.1:8080/html/quickmail_privacy_policy.htm...
My Computer
start Wind... JDev... MailE... D:\D... term ... MyM... 8:07 AM

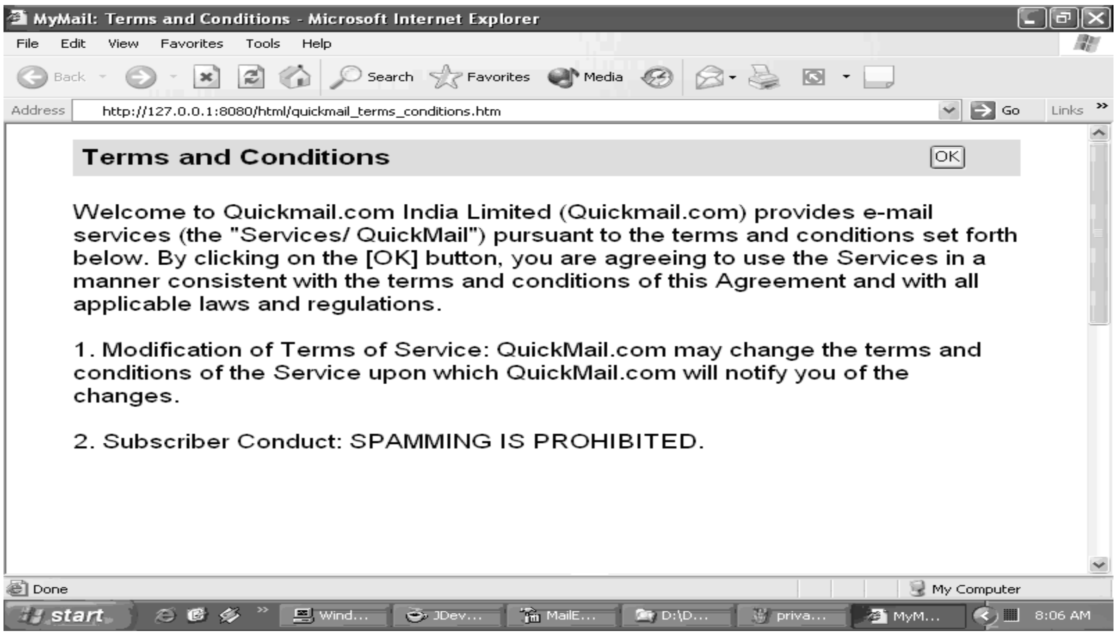
Privacy Policy

This site is the responsibility of QuickMail.com.

We want you to

- feel comfortable using our web sites
- feel secure submitting information to us
- contact us with your questions or concerns about privacy on this site
- know that by using our sites you are consenting to the collection of certain data

9.19 TERMS AND CONDITION PAGE


MyMail: Terms and Conditions - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:8080/html/quickmail_terms_conditions.htm
Go Links
OK
Done
My Computer
start Wind... JDev... MailE... D:\D... priva... MyM... 8:06 AM

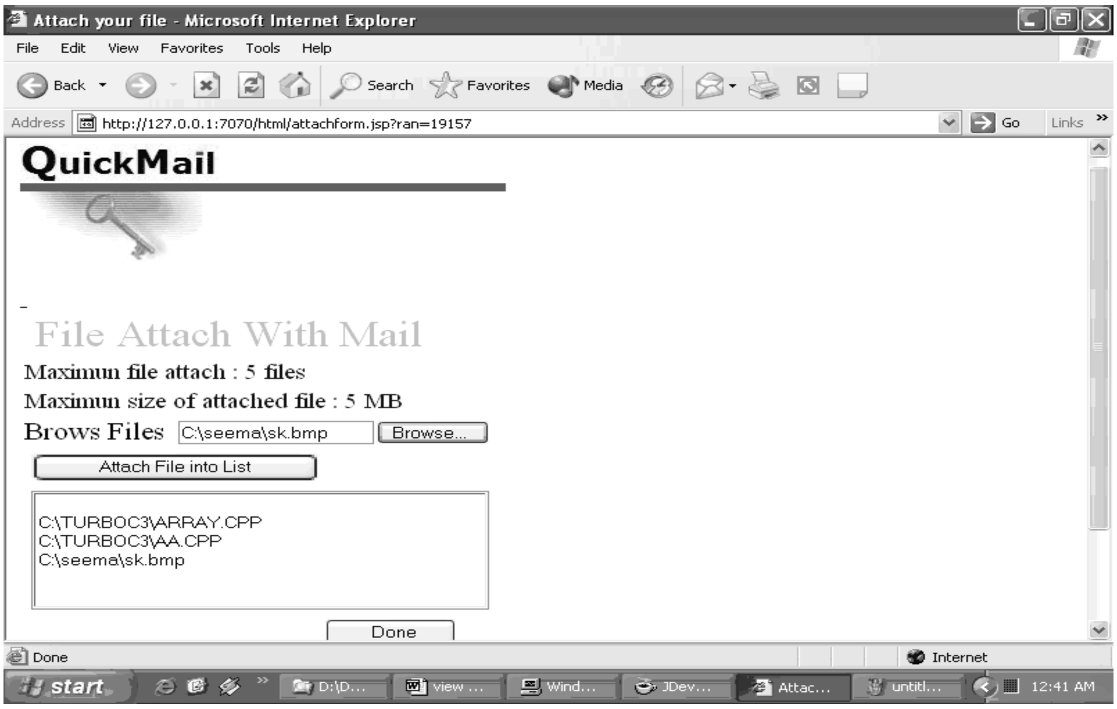
Terms and Conditions

Welcome to Quickmail.com India Limited (Quickmail.com) provides e-mail services (the "Services/ QuickMail") pursuant to the terms and conditions set forth below. By clicking on the [OK] button, you are agreeing to use the Services in a manner consistent with the terms and conditions of this Agreement and with all applicable laws and regulations.

1. Modification of Terms of Service: QuickMail.com may change the terms and conditions of the Service upon which QuickMail.com will notify you of the changes.

2. Subscriber Conduct: SPAMMING IS PROHIBITED.

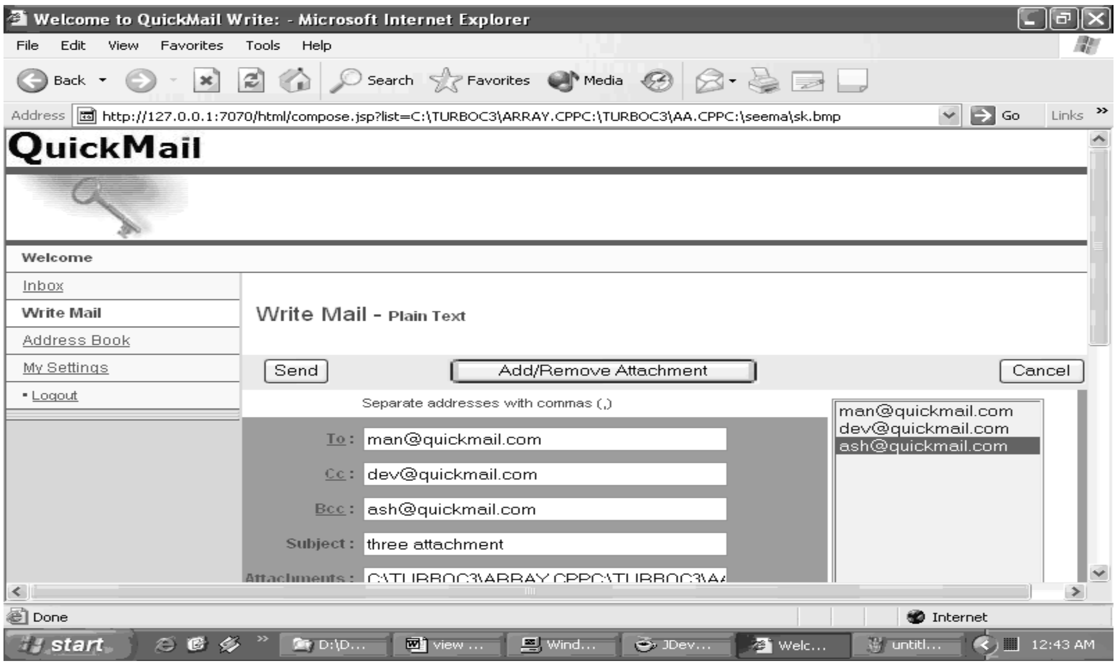
9.20 ATTACH PAGE (EXPERIMENTAL)


Attach your file - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/html/attachform.jsp?ran=19157
QuickMail
File Attach With Mail
Browse...
Attach File into List
Done
Internet
start
12:41 AM

Maximun file attach : 5 files
Maximun size of attached file : 5 MB
Brows Files C:\seema\sk.bmp

C:\TURBOC3\ARRAY.CPP
C:\TURBOC3\AA.CPP
C:\seema\sk.bmp

9.21 COMPOSE PAGE WITH ATTACHMENT (EXPERIMENTAL)


Welcome to QuickMail Write: - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:7070/html/compose.jsp?list=C:\TURBOC3\ARRAY.CPPC:\TURBOC3\AA.CPPC:\seema\sk.bmp
QuickMail
Welcome
Inbox
Write Mail
Address Book
My Settings
Logout
Write Mail - Plain Text
Send
Add/Remove Attachment
Cancel
Internet
start
12:43 AM

Separate addresses with commas (,)

To : man@quickmail.com
Cc : dev@quickmail.com
Bcc : ash@quickmail.com
Subject : three attachment
Attachments : C:\TURBOC3\ARRAY.CPPC:\TURBOC3\A

man@quickmail.com
dev@quickmail.com
ash@quickmail.com

PROGRAM CODE

10 PROGRAM CODE

Program codes of some of the files used in the program are given as follows:

10.1 UserController.java

```
/*

*       This project is based on the MVC (Model View Controller) architecture

*       This is the coding of the controller.

*/


import javax.servlet.*;
import javax.servlet.http.HttpSession.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

import java.sql.*;

//import class
import quickmail.user.login.*;

public class UserController extends HttpServlet {

  private String ctrl="";
  public void init(ServletConfig config) throws ServletException
  {
    super.init(config);
```

```
 }

 public void service(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException
 {
  try
   {

     ctrl=req.getParameter("x");
     System.out.println("test");
   }
   catch(Exception e)
   {
     System.out.println("  l"+ctrl+"  "+e.getMessage());
   }

   if(ctrl.equals("userlogin"))
   {
     System.out.println("test1"+ctrl);
     try
     {
       session=req.getSession(true);
       session.setAttribute("userid",req.getParameter("userid"));
       session.setAttribute("userpass",req.getParameter("userpass"));
       res.sendRedirect("login.jsp");
     }
     catch(Exception e)
     {
       System.out.println("error in the userlogin "+e.getMessage());
     }
   }
   else if(ctrl.equals("user_signup"))
```

```
{
 System.out.println("aaaaaaaa");
 String userid=req.getParameter("userid");
 quickmail.user.login.UserAval aval= new quickmail.user.login.UserAval(userid);
 if(aval.find())
 {
   res.sendRedirect("userexist.jsp?userid="+userid);
  // return;
 }
 else
 {
   String userpass=req.getParameter("userpass");
 //password recall information
 // user selected if own question
   String secretquestion=req.getParameter("secretques");
   if(secretquestion.equals("My own Question"))
     secretquestion=req.getParameter("secretword");

   String cluetoword=req.getParameter("cluetoword");


   String dd=req.getParameter("dd");
   String mm=req.getParameter("mm");
   String yy=req.getParameter("yy");
 //cheacking email is empty
   String email=req.getParameter("email");
   if(email.equals(""))
     email=" ";

   //tell about yu self
   String fname=req.getParameter("fname");
   String lname=req.getParameter("lname");
   String sex=req.getParameter("sex");
   String mstatus=req.getParameter("mstatus");
 //if country is others case handled
 String country=req.getParameter("select_country");
 if(country.equals("Others"))
   country=req.getParameter("country_text");

 //if state is others case handled
 String state=req.getParameter("select_state");
 if(state.equals("Others"))
   state=req.getParameter("state_text");

 //if city is others case handled
 String city=req.getParameter("select_city");
 if(city.equals("Others"))
   city=req.getParameter("city_text");

   String pcode=req.getParameter("pcode");

   //educationd is null or not null
   String edu=req.getParameter("educationd");
   if(edu.equals(""))
     edu=" ";

   //industryd is null or not null
   String indust=req.getParameter("industryd");
   if(indust.equals(""))
     indust=" ";

   //industryd is null or not null
   String occu=req.getParameter("occupationd");
   if(occu.equals(""))
```

```
          occu=" ";
        System.out.println("Before sign up");

        Login log=new Login();
        if(!log.singleRecordFind(quickmail.user.ip.IP.dsn, "login", " uid='"+ userid +
"'"))
        {
          quickmail.user.login.SignUp signup=new quickmail.user.login.SignUp(userid,
userpass, secretquestion, cluetoword, dd, mm, yy, email, fname, lname, sex, mstatus,
country, state, city, pcode, edu, indust, occu);
          int insert=signup.getInsert();
          if(insert > 0)
          {
            res.sendRedirect("acceptcondition.jsp");
          }
          else
          {
            System.out.println("sign up insert val :"+insert);
            res.sendRedirect("problem.jsp");
          }
          HttpSession session=req.getSession();
          session.setAttribute("userid",userid);
          session.setAttribute("userpass",userpass);
        }
        else
        {
          res.sendRedirect("registrationerror.jsp");
        }
      }
    }
    else if(ctrl.equals("filehandle"))
    {
```

```
   String str= req.getParameter("filenm");
   res.sendRedirect("filehandle.jsp?filenm="+str);
  }
  else if(ctrl.equals("Delete"))
  {// it have onlt some values
   try
   {
     String selall=req.getParameter("selectall1");
     String chkbox="";
     String totmsg=req.getParameter("totmsg");
     int tot=Integer.parseInt(totmsg);
     int i=0;
     String arr="undifined";
     while(i < tot)
     {
       chkbox=req.getParameter("SelMsg"+i);
       if(chkbox != null)
         arr+=","+chkbox;
       i++;
     }
     res.sendRedirect("deletemail.jsp?num="+arr+"&mailnum="+tot);
   }
   catch(Exception e)
   {
     e.printStackTrace();
   }

  }
  else if(ctrl.equals("Logout"))
  {// it have onlt some values
   try
   {
       res.sendRedirect("logout.jsp");
       session.invalidate();
   }
   catch(Exception e)
   {
     System.out.println("Error in the logout "+e.getMessage());
   }



  }
  else if(ctrl.equals("addressbookaddnew"))
  {
   String nicnm=req.getParameter("nick");
   String email=req.getParameter("email");
   String userid=req.getParameter("userid");

   String fname=req.getParameter("fname");
     if(fname.equals(""))   fname="a"; else fname=fname.toLowerCase();

   String lname=req.getParameter("lname");
     if(lname.equals(""))   lname="a"; else lname=lname.toLowerCase();

   String cell=req.getParameter("cell");
     if(cell.equals(""))   cell="a"; else cell=cell.toLowerCase();

   String homeph=req.getParameter("homeph");
     if(homeph.equals(""))   homeph="a"; else homeph=homeph.toLowerCase();
   String workph=req.getParameter("workph");
         if(workph.equals(""))   workph="a";
   String fax=req.getParameter("fax");
         if(fax.equals(""))   fax="a";
```

```
String addr=req.getParameter("address");
   if(addr.equals(""))   addr="a"; else addr=addr.toLowerCase();
String city=req.getParameter("city");
   if(city.equals(""))   city="a"; else city=city.toLowerCase();
String state=req.getParameter("state");
   if(state.equals(""))   state="a"; else state=state.toLowerCase();
String pin=req.getParameter("pin");
   if(pin.equals(""))   pin="a"; else pin=pin.toLowerCase();
String country=req.getParameter("country");
   if(country.equals(""))   country="a"; else country=country.toLowerCase();

if(!nicnm.equals(""))   nicnm=nicnm.toLowerCase();
if(!email.equals(""))   email=email.toLowerCase();



if(nicnm.equals("") && email.equals(""))
{
 res.sendRedirect("uncompleteaddressbook.jsp");
}
else
{
 quickmail.user.login.Login login=new quickmail.user.login.Login();
 Statement stmt=login.getStmt();
 try
 {
  if(login.singleRecordFind("ADDRESSBOOK", " email='"+email+"'"))
  {
   res.sendRedirect("existitem.jsp");
  }
  else
  {
```

```
            int insert=stmt.executeUpdate("INSERT INTO ADDRESSBOOK
VALUES('"+userid+"','"+nicnm+"','"+email+"','"+fname+"','"+lname+"','"+cell+"','"+
homeph+"','"+workph+"','"+addr+"','"+city+"','"+state+"','"+pin+"','"+country+"')");
          }
        }
        catch(Exception e)
        {
          System.out.println("error in inserting address book ");
        }

      }
    }
  }

public HttpSession session;
}
```

10.2 IP.java

```
/*

*          This class has system IP and mail name

*         (e.g. 127.0.0.1 & quickmail.com)

*         After perform some changes in this file you can run the program

*        on any computer

*/




package quickmail.user.ip;
// this class have system IP amd mail name(eg 10.0.0.5 & quickmail.com)
public class IP {

  public IP()
{
  }
public static String ip="127.0.0.1";
public static String mail="quickmail.com";
public static String mailatrate="@quickmail.com";
public static String dsn="quickmail";
}
```

10.3 composeprocess.jsp

```
/*
*         This JSP is used in composing of mail
*/




<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<%// page import="javax.mail.session.* "%>
<%@ page import="javax.mail.store.*" %>
<%@ page import="javax.mail.folder.*" %>
<%@ page import="java.util.*" %>
<%@ page import="java.sql.*" %>
<%@ page import="javax.mail.*" %>
<%@ page import="javax.mail.internet.*" %>
<%@ page import="javax.activation.*" %>

<%! String userid, userpass; %>
<%! String to1,cc,bcc,attach,msgtxt, subject,recept,signature,sendmsg; %>
<%
// creating an object of Login.class
quickmail.user.login.Login login= new quickmail.user.login.Login();

  userid=(String)session.getAttribute("userid");
  userpass= (String)session.getAttribute("userpass");
%>
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=WINDOWS-1252">
<META content="Microsoft FrontPage 4.0" name=GENERATOR></HEAD>
```

```
<LINK href="../images/styles.css" type=text/css
rel=stylesheet><LINK
href="../images/register.css" rel=stylesheet>

<TITLE>
  sending message process:
</TITLE>
</HEAD>
<BODY leftMargin=0 topMargin=0 marginheight="0" marginwidth="0">



<TABLE cellSpacing=0 cellPadding=0 width="100%"
background="../images/pobtrans.gif" border=0 dwcopytype="CopyTableRow">
  <TR>
    <TD align=left>
    <IMG
      alt="QuickMail" src="../images/LogoBig.jpg" border=0 width="143"
height="34"></TD></TR>
  <TR>
    <TD bgcolor="#336699" height="7">
    <IMG height=7 border=0 width="7"></TD></TR>
</TABLE>

<%
  to1       =request.getParameter("To");
  cc        =request.getParameter("Cc");
  bcc       =request.getParameter("Bcc");
  attach    =request.getParameter("list");
  msgtxt    =request.getParameter("msg_text");
  subject   =request.getParameter("subject");
  recept    =request.getParameter("return_recp");
  if(recept==null)
      recept="1";

  signature =request.getParameter("signature1");
  if(signature==null)
      signature="1";

  sendmsg   =request.getParameter("sent_messages1");
%>

<%

try
{
  boolean sessionDebug=false;

  Properties p=new Properties();
  p.put("mail.smtp.host",quickmail.user.ip.IP.ip);
  Session s=Session.getInstance(p,null);
  MimeMessage m=new MimeMessage(s);

  InternetAddress from =new
InternetAddress(userid+quickmail.user.ip.IP.mailatrate);//"fareed@quickmail.com");

  String local="";
  StringTokenizer st= new StringTokenizer(to1,",");
  while(st.hasMoreTokens())
  {
    local=st.nextToken();
    InternetAddress to= new InternetAddress(local);
    m.addRecipient(m.RecipientType.TO,to);
  }
```

```
System.out.println("asasasasas");

  m.setSentDate(new java.util.Date());
  m.setFrom(from);

//if user checks at return recipt
  if(recept.equals("1")!=true)
  {
      InternetAddress recp= new
InternetAddress(userid+quickmail.user.ip.IP.mailatrate);
      m.addRecipient(m.RecipientType.CC,recp);
  }
  // add signature
System.out.println("22222222222222222222");
  if(!signature.equals("1"))
  {
      msgtxt+=userid;
  }

  StringTokenizer stcc= new StringTokenizer(cc,",");
  if(!cc.equals(""))
  {
      while(stcc.hasMoreTokens())
      {
        local=stcc.nextToken();
        InternetAddress ccc= new InternetAddress(cc);
        m.addRecipient(m.RecipientType.CC,ccc);
      }
  }

  StringTokenizer stbcc= new StringTokenizer(cc,",");
  if(!bcc.equals(""))
```

```
 {
   while(stbcc.hasMoreTokens())
    {
     local=stbcc.nextToken();
     InternetAddress bccc= new InternetAddress(bcc);
     m.addRecipient(m.RecipientType.BCC,bccc);
    }
 }

 m.setSubject(subject);

//create the first part
  Multipart mailBody=new MimeMultipart();
  MimeBodyPart mainBody=new MimeBodyPart();
  mainBody.setText(msgtxt);
  mailBody.addBodyPart(mainBody);

  System.out.println("3333333333333333333"+attach);
if(attach.equals(" -- no attachments -- "))
{
 System.out.println("in hhhhhhhhhhhhhhhhhh  "+attach);
        //create the second part with the attachment
 StringTokenizer stattach= new StringTokenizer(attach,",");

 while(stattach.hasMoreTokens())
 {
         FileDataSource fds=new FileDataSource(stattach.nextToken());
         MimeBodyPart mimeAttach=new MimeBodyPart();
         mimeAttach.setDataHandler(new DataHandler(fds));
         mimeAttach.setFileName(fds.getName());
         mailBody.addBodyPart(mimeAttach);
 }
```

```
        //create the body of the mail

}
   m.setContent(mailBody);
   Transport.send(m);
System.out.println("44444444444444444444444");
// if msg successfully send
%>
&nbsp;
<b></b>
<table border="0" width="100%" bgcolor="#f7f7e7">
  <tr>
    <td width="100%" colspan="3" class=heading1><font color="#5E98F0"
size="5">Mail send successfully</font></td>
  </tr>
  <tr>
    <td width="25%" rowspan="3"><a href="compose.jsp">Write another
mail</a></td>
    <td width="25%">To</td>
    <td width="50%"><%=to1%></td>
  </tr>
  <tr>
    <td width="25%">Cc</td>
    <td width="50%"><%=cc%></td>
  </tr>
  <tr>
    <td width="25%" rowspan="2">Bcc</td>
    <td width="50%" rowspan="2"><%=bcc%></td>
  </tr>
  <tr>
    <td width="25%" rowspan="3"><a href="inbox.jsp">Inbox</a></td>
  </tr>
  <tr>
    <td width="25%">Subject</td>
    <td width="50%"><%=subject%></td>
  </tr>
  <tr>
    <td width="25%">Attachment</td>
    <td width="50%"><%=attach%></td>
  </tr>
</table>

<%
}
catch(Exception e)
{
  System.out.println("error in composeprocess.jsp "+e.getMessage());
%>
&nbsp;<b></b>
<b></b>
<table border="0" width="100%" bgcolor="#f7f7e7">
  <tr>
    <td width="100%" colspan="3"><font size="4" color="#FF0000">Error find at
      that time it cannot send the mail</font></td>
  </tr>
  <tr>
    <td width="25%" rowspan="2"><a href="compose.jsp">Write another
mail</a></td>
    <td width="25%">To</td>
    <td width="50%">&nbsp;<%=to1%></td>
  </tr>
  <tr>
    <td width="25%" rowspan="2">Cc</td>
    <td width="50%" rowspan="2">&nbsp;<%=cc%></td>
```

```
 </tr>
 <tr>
   <td width="25%" rowspan="2"><a href="sdsd">Add new Address Book</a></td>
 </tr>
 <tr>
   <td width="25%" rowspan="2">Bcc</td>
   <td width="50%" rowspan="2">&nbsp;<%=bcc%></td>
 </tr>
 <tr>
   <td width="25%" rowspan="3"><a href="inbox.jsp">Inbox</a></td>
 </tr>
 <tr>
   <td width="25%">Subject</td>
   <td width="50%">&nbsp;<%=subject%></td>
 </tr>
 <tr>
   <td width="25%">Attachment</td>
   <td width="50%">&nbsp;<%=attach%></td>
 </tr>
</table>

<%
}
%>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
 <TBODY>
 <TR>
   <TD align=left colSpan=2 height=1 bgcolor="#336699">
```

```
  <IMG height=1
    src="sample/Sign%20Ink_files/pobtrans.gif" width="100%"></TD></TR>
 <TR>
  <TD vAlign=top align=left><font face="Arial" size="1">Copyright © 2005 <a
href="http://www.quickmailmail.com/" target="TOP">QuickMail.com</a> India
Limited.
All Rights Reserved. </font></TD>
  <TD vAlign=top align=right><FONT face=Arial size=-2>
  <a href="contact_us.html">Contact Us</a> •
  <a href="quickmailmail_terms_conditions.htm">Terms and Conditions
</a>&nbsp;•
  <a href="quickmail_privacy_policy.htm">Privacy
    Policy</a></FONT></TD></TR>
 <TR>
  <TD vAlign=top align=right colSpan=2><FONT face=Arial size=1>This page is
    managed by MySql</FONT></TD></TR></TBODY></TABLE>
</BODY>
</HTML>
<%
//------TO-----------------------
// new email id insert into addressbook after sending

try
{
 Statement stmt=login.getStmt(quickmail.user.ip.IP.dsn);
 StringTokenizer st1to= new StringTokenizer(to1,",");
 int iii=0;
 String local="";
 boolean b=false;
 while(st1to.hasMoreTokens())
 {
     System.out.flush();
```

```
        local=st1to.nextToken();
        b=login.singleRecordFind(quickmail.user.ip.IP.dsn, "addressbook" , "
userid='"+userid+"' and emailid='"+local+"'");
        if(b==true)
        {
          System.out.println("NOT INSERTED loop :>> "+iii+"    "+local);
        }
        else
        {
          int ii=stmt.executeUpdate("insert into addressbook
values('"+userid+"','"+local+"','"+local+"','a','a','a','a','a','a','a','a','a','a')");
          System.out.println("loop :>> "+iii+"   ii :>> "+ii+" insert : "+local);
        }
        iii++;
        local="";
        b=false;
  }
}
catch(Exception e)
{
  System.out.println("to inserting data in addressbook "+e.getMessage());
}
%>


<%
//--------Cc----------------
// new email id insert into addressbook after sending


try
{
  Statement stmt=login.getStmt(quickmail.user.ip.IP.dsn);
  StringTokenizer st1to= new StringTokenizer(cc,",");
  int iii=-1;
  String local="";
  while(st1to.hasMoreTokens())
  {
    local=st1to.nextToken();
    {
      iii=-1;
      ResultSet rs=stmt.executeQuery("Select * from addressbook where
userid='"+userid+"' and emailid='"+local+"'");
      {
        while(rs.next())    iii++;
        if(iii < 0)
        {
          stmt.executeUpdate("insert into addressbook
values('"+userid+"','"+local+"','"+local+"','a','a','a','a','a','a','a','a','a','a')");
          login.commitrec();
        }
      }
    }
    System.out.println("iii :"+iii+" insert : "+local);
  }
}
catch(Exception e)
{System.out.println("to inserting data in addressbook "+e.getMessage());
}
%>
```

```
<%
//--------BCc----------------
// new email id insert into addressbook after sending
try
{
  Statement stmt=login.getStmt(quickmail.user.ip.IP.dsn);
  StringTokenizer st1to= new StringTokenizer(bcc,",");
  int iii=-1;
  String local="";
  while(st1to.hasMoreTokens())
  {
    local=st1to.nextToken();
    {
      iii=-1;
      ResultSet rs=stmt.executeQuery("Select * from addressbook where
userid='"+userid+"' and emailid='"+local+"'");
      {
        while(rs.next())    iii++;
        if(iii < 0)
        {
          stmt.executeUpdate("insert into addressbook
values('"+userid+"','"+local+"','"+local+"','a','a','a','a','a','a','a','a','a','a')");
          login.commitrec();
        }
      }
    }
    System.out.println("iii :"+iii+" insert : "+local);
  }
}
catch(Exception e)
{System.out.println("to inserting data in addressbook "+e.getMessage());
}
```

```
%>
```

10.4 FileTest.java

```
/*
*       This file is used to check the file attached
*       It is still experimental and does
*       not always works
*/
package quickmail.user.login.attach;
import java.io.*;
public class FileTest extends Object {
  public FileTest()
  {
  }
  public static String fileCheck(String filenm)
  {
    String str="";
    try
    {
      File file = new File(filenm);
      if(file.isFile() && file.exists() )
        str="ok";
    }
    catch(Exception e)
    {
      System.out.println("error in FileTest.java :>> "+e.getMessage());
      str="error";
    }
    return str;
  }
}
```

10.5 Login.java

```
/*
*         This file is used in the login process
*         It connects with database
*/

package quickmail.user.login;
import java.sql.*;

public class Login extends Object {

  /** Constructor   */
  public Login() {
  }
  public void commitrec()
  {
    try
    {
      cn.commit();
    }
    catch(Exception e)
    {
      System.out.println("Login.java error : "+ e.getMessage());
    }
  }
  public void setUserid(String uid)
  {
    userid=uid;
```

```
}
public void setUserpass(String upass)
{
  userpass=upass;
}
public Statement getStmt(String dsn)
{
  Connection cn=null;
  Statement stmt=null;
  try
  {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    cn=DriverManager.getConnection("jdbc:odbc:"+dsn);
    stmt = cn.createStatement();
  }
  catch(Exception e)
  {
    System.out.println("Err msg :>> "+e.getMessage());
    stmt=null;
  }
  return stmt;
}
public Statement getStmt()
{
  Connection cn=null;
  Statement stmt=null;
  try
  {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    cn=DriverManager.getConnection("jdbc:odbc:"+dsn);
    stmt = cn.createStatement();
  }
```

```
    catch(Exception e)
    {
      System.out.println("Err msg :>> "+e.getMessage());
      stmt=null;
    }
    return stmt;
  }


  public boolean executeSelectQuery(String dsn,String tabnm)
  {
     ResultSet rs=null;
     int rec=-1;
    try
    {
      rs=getStmt(dsn).executeQuery("SELECT  *  FROM "+tabnm+ "  WHERE
uid='"+userid+"' and upassword='"+userpass+"'");

      while(rs.next())
      {
        rec++;
       // System.out.println("\t"+rs.getString("uid")+"\t"+rs.getString("upassword"));
      }
    }
    catch(Exception e)
    {
      System.out.println("Err msg in getStmt :>> "+e.getMessage());
    }
    if(rec==0)
      return true;
    else
      return false;
```

```
}

public boolean executeSelectQuery(String dsn,String tabnm,String condi)
{
   ResultSet rs=null;
   Statement stmt=null;
   int rec=-1;
  try
  {
    stmt=getStmt(dsn);
    rs=stmt.executeQuery("SELECT  *  FROM "+tabnm+ "  WHERE " +condi);
    while(rs.next())
      rec++;
  }
  catch(Exception e)
  {
    System.out.println("Err msg in getStmt :>> "+e.getMessage());
  }
  if(rec >=0)
  {
    System.out.println("Login.java :>> "+rec);
    return true;
  }
  else
    return false;
}


public boolean singleRecordFind(String dsn, String tabnm)
{
  return executeSelectQuery(dsn,tabnm);
}
```

```
 public boolean singleRecordFind(String dsn, String tabnm, String condi)
 {
   return executeSelectQuery(dsn,tabnm,condi);
 }
 public String getUserid()
 {
   return userid;
 }

 public String getUserpass()
 {
   return userpass;
 }

   private  String userid="", userpass="";
   private Connection cn=null;
   private Statement stmt=null;
   private ResultSet rs=null;
   private String dsn=quickmail.user.ip.IP.dsn;
}
```

10.6 <u>SignUp.java</u>

```
/*
*        This java file is ued in the sign up (registration) process
*        It saves the user info in database
*/


package quickmail.user.login;
import java.sql.*;

public class SignUp extends Object
{

  public  SignUp( String userid,String  userpass,String  secretquestion,String
cluetoword,String  dd,String  mm,String  yy,String  email,String  fname,String
lname,String  sex,String  mstatus,String  country,String  state,String  city,String
pcode,String  edu,String  indust,String  occu)
  {
    this.userid=userid;
    this.userpass=userpass;
    this.secretques=secretquestion;
    this.cluetoword=cluetoword;
    this.dd=dd;
    this.mm=mm;
    this.yy=yy;
    this.email=email;
    this.fname=fname;
    this.lname=lname;
    this.sex=sex;
    this.mstatus=mstatus;
    this.country=country;
    this.state=state;
    this.pcode=pcode;
    this.educationd=edu;
    this.industryd=indust;
    this.occupationd=occu;
  }

  public void setUserid(String uid)
  {
    userid=uid;
  }
  public void setUserpass(String upass)
  {
    userpass=upass;
  }
  public int getInsert()
  {

    int insert=-1;
    int insert1=-1;
    int insert2=-1;
    Connection cn=null;
    Statement pstmt=null;

  //  PreparedStatement pstmt1=null;
    try
    {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      cn=DriverManager.getConnection("jdbc:odbc:"+dsn);
```

```
   pstmt = cn.createStatement();
   insert=pstmt.executeUpdate("insert into LOGIN
values('"+userid+"','"+userpass+"')");

   insert1=pstmt.executeUpdate("insert into PWDRECALL
values('"+secretques+"','"+cluetoword+"','"+mm+"/"+dd+"/"+yy+"','"+email+"','"+use
rid+"')");

   insert2=pstmt.executeUpdate("insert into USER_INFO
values('"+userid+"',"+fname+"','"+lname+"','"+sex+"','"+mstatus+"','"+country+"','"+s
tate+"','"+pcode+"','"+industryd+"','"+occupationd+"','"+educationd+"')");
   insert=insert+insert1+insert2;
   if(insert > 2)
   {
    System.out.println("in signup Commited");
    cn.commit();
   }
   else
   {
    System.out.println("in signup Rollbacked");
    cn.rollback();
   }

  }
  catch(Exception e)
  {
   System.out.println("Err msg :>> "+e.getMessage());
   pstmt=null;
  }
  System.out.println("asas  "+ insert);
  return insert;
 }
```

```
  public String getUserid()
  {
    return userid;
  }

  public String getUserpass()
  {
    return userpass;
  }



    private String dsn="quickmail";
    private  String userid="", userpass="";
    private  String secretques="", cluetoword="";
    private  String dd="", mm="", yy="";
    private  String email="", fname="";
    private  String lname="", sex="";

    private  String mstatus="", country="";
    private  String state="", city="";
    private  String pcode="", educationd="";
    private  String industryd="", occupationd="";



    private Connection cn=null;
    private Statement stmt=null;
}
```

10.7 UserAval.java

```
/*
*         This Java file checks the availability of the
*         user name entered and responds with the
*         appropriate answer
*/



package quickmail.user.login;
import java.sql.*;

public class UserAval extends Object {

  public UserAval(String userid)
  {
    this.userid=userid;
  }



  public Statement getStmt()
  {
    Connection cn=null;
    Statement stmt=null;
    try
    {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      cn=DriverManager.getConnection("jdbc:odbc:"+dsn);
      stmt = cn.createStatement();
    }
    catch(Exception e)
    {
      System.out.println("Err msg :>> "+e.getMessage());
      stmt=null;
    }
    return stmt;
  }

  public boolean executeSelectQuery()
  {
     ResultSet rs=null;
     int rec=-1;
    try
    {
      rs=getStmt().executeQuery("SELECT  *  FROM LOGIN WHERE
uid='"+userid+"'");
      while(rs.next())
      {
        rec++;
        System.out.println("\t"+rs.getString("uid")+"\t"+rs.getString("upassword"));
      }
    }
    catch(Exception e)
    {
      System.out.println("Err msg in getStmt :>> "+e.getMessage());
    }
    if(rec==0)
      return true;
    else
      return false;
  }
```

```
 public boolean find()
 {
   return executeSelectQuery();
 }

 public String getUserid()
 {
   return userid;
 }

 String dsn="quickmail";
   private Connection cn=null;
   private Statement stmt=null;
   private ResultSet rs=null;
   String userid="";
}
```

10.8 <u>userfind.jsp</u>

```
/*
*         This jsp file displays the error if the
*         userid is not available and asks to
*         use some other user name
*/


<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<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>Continue with this page</title>
</head>
<body>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <tr>
    <TD align=left width="100%">
    <IMG
      alt="QuickMail" src="../images/LogoBig.jpg" border=0 width="143"
height="34"></TD>
  </tr>
  <tr>
    <TD bgcolor="#336699" height="7" width="528">
    <IMG height=7 border=0 width="7"></TD>
  </tr>
  <tr>
    <TD vAlign=top width="100%">
    <a href="../images/clogin.jpg"><IMG src="../images/clogin.jpg" border=0
    valign="TOP" align="left"></a></TD>
  </tr>
  <tr>
    <TD bgColor="#336699" height=5 width="528">
      <TABLE cellSpacing=0 cellPadding=0 border=0 bgcolor="#336699">
        <TBODY>
        <TR>
          <TD height=1 bgcolor="#336699"></TD></TR>
</TBODY>
</TABLE>
</TD>
  </tr>
  </TBODY>
</TABLE>

<table border="0" width="679" bgcolor="#BBC7E3">
  <tr>
    <td width="341"><font face="Arial" size="3"><b>User ID</b></font></td>
    <td width="324"><font face="Arial" size="4"
color="#0000FF"><b><%=request.getParameter("userid")%>
</b></font></td>
  </tr>
  <tr>
    <td width="341"><font face="Arial" size="2"><b><b>is not available at the
quickmail.com</b>
</font>
</td>
```

```
  </tr>
  <tr>
    <td width="671" colspan="3"><font face="Arial" size="3"><a
href="javascript:window.close()">Close </a> Try with other QuickMail userID
</font>
</td>
</tr>
</table>

</body>
</html>
```

10.9 usernotfind.jsp

```
/*
*	This jsp file displays the availability of the
*	userid and asks the user to use it.
*/



<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<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>Continue with this page</title>
</head>

<body>

<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <tr>
    <TD align=left width="100%">
    <IMG
      alt="QuickMail" src="../images/LogoBig.jpg" border=0 width="143"
height="34"></TD>
  </tr>
```

```
 <tr>
   <TD bgcolor="#336699" height="7" width="528">
   <IMG height=7 border=0 width="7"></TD>
 </tr>
 <tr>
   <TD vAlign=top width="100%">
   <a href="../images/clogin.jpg"><IMG src="../images/clogin.jpg" border=0
   valign="TOP" align="left"></a></TD>
 </tr>
 <tr>
   <TD bgColor="#336699" height=5 width="528">
     <TABLE cellSpacing=0 cellPadding=0 border=0 bgcolor="#336699">
       <TBODY>
       <TR>
         <TD height=1 bgcolor="#336699"></TD>
</TR></TBODY>
</TABLE>
</TD>
 </tr>
 </TBODY>
</TABLE>
<table border="0" width="679" bgcolor="#BBC7E3">
 <tr>
   <td width="341"><font face="Arial" size="3"><b>User ID</b></font></td>
   <td width="324"><font face="Arial" size="4"
color="#0000FF"><b><%=request.getParameter("userid")%></b></font></td>
 </tr>
 <tr>
   <td width="341"><font face="Arial" size="2"><b>is&nbsp; available at the
quickmail.com</b></font>
</td>
 </tr>
```

```
  <tr>
    <td width="671" colspan="3"><font face="Arial" size="3"><a
href="javascript:window.close()">Close </a>this window and Enjoy with this
ID</font></td>
  </tr>
</table>
</body>
</html>
```

10.10 step1.jsp

```
/*
*       This jsp is the first step
*       in the password recovery process.
*       It asks the user to enter some
*       personal details.
*/


<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<html><head><title>Quick: Forgot your Password?</title>

<script language="JavaScript">

 function winclose()
   {
     window.close()
   }

</script></head>
<body alink="#0066ff" bgcolor="#ffffff" link="#0066ff" text="#000000"
vlink="#0066ff">
<form method="get" action="step11.jsp">
<input name="FormName" value="showhqa" type="hidden">
<center>
<table bgcolor="#dfdfdf" border="0" cellpadding="1" cellspacing="0"
width="448"><tbody><tr><td align="center" width="458">
<table border="0" cellpadding="0" cellspacing="0" width="446">
```

```
<tbody><tr bgcolor="#dfdfdf">
<td height="40" valign="middle" width="15">&nbsp;</td>
<td valign="middle" width="378">
<font color="#000000" face="arial" size="+1"><b>Forgot your
Password?</b></font>
</td>
<td align="center" valign="middle" width="53">
<input name="btn_name" value="OK" onclick="window.close()" type="button">
</td>
</tr>
<tr bgcolor="#ffffff">
<td colspan="3" height="5" width="456"><font size="-2">&nbsp;</font><br>
</td>
</tr>
<tr bgcolor="#ffffff">
<td height="40" valign="middle" width="15">&nbsp;</td>
<td colspan="2" valign="middle" width="431">
<font color="#000000" face="arial" size="-1">
Just follow these two steps to get a New Password.<br><br>Note: This process asks
you for the answer to your hint question.
<br><br>
<b>Step 1:</b> Please enter your personal details below.
</font>
<br>
<table border="0" cellpadding="0" cellspacing="0" width="430">
<tbody><tr>
<td colspan="2" align="right" height="10" valign="middle" width="430">
<font color="#000000" face="Arial" size="-2">&nbsp;</font><br>
</td>
</tr>

<tr>
```

```
<td align="right" height="30" valign="middle" width="145">
<font color="#000000" face="Arial" size="-1">
Username&nbsp;&nbsp;&nbsp;
</font>
</td>
<td valign="middle" width="385">
<input name="login" size="15" type="text"><br>
</td>
</tr>

<tr>
<td align="right" height="30" valign="middle" width="145">
<font color="#000000" face="Arial" size="-1">
Date of Birth&nbsp;&nbsp;&nbsp;
</font>
</td>
<td valign="middle" width="385">
        <select name="month"><option value="0">Month
        </option><option value="1">Jan
        </option><option value="2">Feb
        </option><option value="3">Mar
        </option><option value="4">Apr
        </option><option value="5">May
        </option><option value="6">Jun
        </option><option value="7">Jul
        </option><option value="8">Aug
        </option><option value="9">Sep
        </option><option value="10">Oct
        </option><option value="11">Nov
        </option><option value="12">Dec
        </option></select>        <select name="day"><option value="0">Day
        </option><option value="01">01
```

```
</option><option value="02">02
</option><option value="03">03
</option><option value="04">04
</option><option value="05">05
</option><option value="06">06
</option><option value="07">07
</option><option value="08">08
</option><option value="09">09
</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>          <select name="year"><option value="0">Year
</option><option value="2001">2001
```

```
</option><option value="2000">2000
</option><option value="1999">1999
</option><option value="1998">1998
</option><option value="1997">1997
</option><option value="1996">1996
</option><option value="1995">1995
</option><option value="1994">1994
</option><option value="1993">1993
</option><option value="1992">1992
</option><option value="1991">1991
</option><option value="1990">1990
</option><option value="1989">1989
</option><option value="1988">1988
</option><option value="1987">1987
</option><option value="1986">1986
</option><option value="1985">1985
</option><option value="1984">1984
</option><option value="1983">1983
</option><option value="1982">1982
</option><option value="1981">1981
</option><option value="1980">1980
</option><option value="1979">1979
</option><option value="1978">1978
</option><option value="1977">1977
</option><option value="1976">1976
</option><option value="1975">1975
</option><option value="1974">1974
</option><option value="1973">1973
</option><option value="1972">1972
</option><option value="1971">1971
</option><option value="1970">1970
</option><option value="1969">1969
```

```
          </option><option value="1968">1968
          </option><option value="1967">1967
          </option><option value="1966">1966
          </option><option value="1965">1965
          </option><option value="1964">1964
          </option><option value="1963">1963
          </option><option value="1962">1962
          </option><option value="1961">1961
          </option><option value="1960">1960
          </option><option value="1959">1959
          </option><option value="1958">1958
          </option><option value="1957">1957
          </option><option value="1956">1956
          </option><option value="1955">1955
          </option><option value="1954">1954
          </option><option value="1953">1953
          </option><option value="1952">1952
          </option><option value="1951">1951
          </option><option value="1950">1950
          </option><option value="1949">1949
          </option><option value="1948">1948
          </option><option value="1947">1947
          </option><option value="1946">1946
          </option><option value="1945">1945
          </option><option value="1944">1944
          </option></select><br>
</td></tr>
<tr><td align="right" height="30" valign="middle" width="145">
<font color="#000000" face="Arial" size="-1">
City&nbsp;&nbsp;&nbsp;
</font>
</td>
```

```
<td valign="middle" width="385">
<input tyep="text" name="city" size="15"><br>
</td>
</tr>
<tr>
<td align="right" height="30" valign="middle" width="145">
<font color="#000000" face="Arial" size="-1">
Country&nbsp;&nbsp;&nbsp;
</font>
</td>
<td valign="middle" width="385">
                <select name="country"><option value="99"
selected="selected">India
                </option><option value="222">United States
                </option><option value="221">United Kingdom
                </option><option value="37">Canada
                </option><option value="189">Singapore
                </option><option value="128">Malaysia
                </option><option value="12">Australia
                </option><option value="185">Saudia Arabia
                </option><option value="220">United Arab Emirates
                </option><option value="194">South Africa
                </option><option value="238">Others
                </option></select>
</td>
</tr>
<tr>
<td align="right" height="30" valign="middle" width="145">
<font color="#000000" face="Arial" size="-1">
&nbsp;<br>
</font>
</td>
```

```
<td valign="middle" width="385"><br>
<input value="Please Continue" type="submit">
</td>
</tr>
</tbody></table>
<br><br><br><br>
</td>
</tr>
<tr bgcolor="#eeeeee">
<td height="40" valign="middle" width="15">&nbsp;</td>
<td valign="middle" width="378">
<font face="arial" size="-1">
<b>
<font color="red">Important:</font></b> This information should be the same as that
provided<br>
when you created your email account.</font><br>
</td>
<td align="center" valign="middle" width="53">
&nbsp;
</td></tr>
</tbody></table>
</td></tr>
</tbody></table>
<br>
<font color="#000000" face="ms sans serif" size="-2">
<center><font color="#000000" face="ms sans serif" size="1">Copyright © 2005 <a
href="http://www.quickmail.com/" target="TOP">QuickMail.com</a> India Limited.
All Rights Reserved.
</center></font>
</center>
</form>
</body></html>
```

10.11 step2.jsp

```
/*
*       This jsp is the second step
*       in the password recovery process.
*       It asks the user to enter
*       the secret question and
*       secret answer.
*/


<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<html><head><title>QuickMail: Forgot your Password?</title>

<script language="JavaScript">

 function winclose()
   {
     window.close()
   }


</script></head>
<body alink="#0066ff" bgcolor="#ffffff" link="#0066ff" text="#000000"
vlink="#0066ff">
<form method="get" action="/bn/passwd_remind.cgi">
<input name="FormName" value="showhqa" type="hidden">
```

```
<center>
<table bgcolor="#dfdfdf" border="0" cellpadding="1" cellspacing="0"
width="448"><tbody><tr><td align="center" width="458">
<table border="0" cellpadding="0" cellspacing="0" width="446">
<tbody><tr bgcolor="#dfdfdf">
<td height="40" valign="middle" width="15">&nbsp;</td>
<td valign="middle" width="378">
<b><font face="arial" size="+1">Answer</font></b><font color="#000000"
face="arial" size="+1"><b>
your secret Question</b></font><font color="#000000">.</font></td>
<td align="center" valign="middle" width="53">
<input name="btn_name" value="OK" onclick="window.close()" type="button">
</td>
</tr>
<tr bgcolor="#ffffff">
<td colspan="3" height="5" width="456"><font size="-2">&nbsp;</font><br>
</td>
</tr>
<tr bgcolor="#ffffff">
<td height="40" valign="middle" width="15">&nbsp;</td>
<td colspan="2" valign="middle" width="431">
<font color="#000000" face="arial" size="-1">
Just complete this second step to get a New Password.<br><br>
<b>Step 2:</b> Please answer your hint question.
</font>
<br>
<table border="0" cellpadding="0" cellspacing="0" width="430" height="194">
<tbody><tr>
<td colspan="2" align="right" height="28" valign="middle" width="430">
<font color="#000000" face="Arial" size="-2">&nbsp;</font><br>
</td>
</tr>
```

```
<tr>
<td align="right" height="30" valign="middle" width="145">
<font face="Arial" size="-1">Question&nbsp;&nbsp;&nbsp; </font>
</td>
<td valign="middle" width="385" height="30">
<input name="login" size="41" type="text"><br>
</td>
</tr>
<tr>
<td align="right" height="30" valign="middle" width="145">
&nbsp;</td>
<td valign="middle" width="385" height="30">
            &nbsp;</td>
</tr>
<tr>
<td align="right" height="30" valign="middle" width="145">
<font face="Arial" size="-1">Answer&nbsp;&nbsp;&nbsp; </font>
</td>
<td valign="middle" width="385" height="30">
<input tyep="text" name="city" size="22"><br>
</td>
</tr>
<tr>
<td align="right" height="30" valign="middle" width="145">
&nbsp;</td>
<td valign="middle" width="385" height="30">
                  &nbsp;</td>
</tr>
<tr>
<td align="right" height="46" valign="middle" width="145">
<font color="#000000" face="Arial" size="-1">
&nbsp;<br>
```

```
</font>
</td>
<td valign="middle" width="385" height="46"><br>
<input value="Please Continue" type="submit">
</td>
</tr>
</tbody>
</table>
<br>
<br>
</td>
</tr>
<tr bgcolor="#eeeeee">
<td height="40" valign="middle" width="15">&nbsp;</td>
<td valign="middle" width="378">
<font face="arial" size="-1">
<font color="red"><b>Important</b>:</font> This answer should be the same as that
provided<br>
when you created your email account.</font><br>
</td>
<td align="center" valign="middle" width="53">
&nbsp;
</td>
</tr>
</tbody>
</table>
</td>
</tr>
</tbody>
</table>
<br>
<font color="#000000" face="ms sans serif" size="-2">
```

```
<center><font color="#000000" face="ms sans serif" size="1">Copyright © 2005 <a
href="http://www.quickmail.com/" target="TOP">QuickMail.com</a> India Limited.
All Rights Reserved.
</center></font>
</center>
</form>
</body></html>
```

10.12 <u>step3.jsp</u>

```
/*
*         This jsp is the third step
*         in the password recovery process.
*         It resets the password with a random
*         password and shows it to the user.
*/



<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<html><head><title>QuickMail: Forgot your Password?</title>
<script language="JavaScript">

 function winclose()
   {
     window.close()
   }

</script></head>
<body alink="#0066ff" bgcolor="#ffffff" link="#0066ff" text="#000000"
vlink="#0066ff">
<form >
<input name="FormName" value="showhqa" type="hidden">
<center>
<table bgcolor="#dfdfdf" border="0" cellpadding="1" cellspacing="0"
width="448"><tbody><tr><td align="center" width="458">
<table border="0" cellpadding="0" cellspacing="0" width="446">
<tbody><tr bgcolor="#dfdfdf">
<td height="40" valign="middle" width="15">&nbsp;</td>
```

```
<td valign="middle" width="374">
<font color="#000000" face="arial" size="+1"><b>This is your New
Password.</b></font>
</td>
<td align="center" valign="middle" width="57">
<input name="btn_name" value="OK" onclick="window.close()" type="button">
</td>
</tr>
<tr bgcolor="#ffffff">
<td colspan="3" height="5" width="456"><font size="-2">&nbsp;</font><br>
</td>
</tr>
<tr bgcolor="#ffffff">
<td height="40" valign="middle" width="15">&nbsp;</td>
<td colspan="2" valign="middle" width="431">
<font color="#000000" face="arial" size="-1">
<b>Step 3:</b>
</font>
<font face="arial" size="-1">This is your new random generated
password.</font><font color="#000000" face="arial" size="-1"><br><br>
</font>
<br>
<table border="0" cellpadding="0" cellspacing="0" width="433" height="85">
<tbody><tr>
<td align="center" height="37" width="433">
<font color="#000000" face="Arial" size="-2">&nbsp;</font><font face="Arial"
size="-1">Password&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
</font>
<input name="login" size="19" type="text"><br>
</td>
</tr>
```

```
<tr>
<td align="center" height="63" width="533">
<font color="#000000" face="Arial" size="-1">
&nbsp;<br>
</font>
<input value="Continue to Login" type="submit">
</td>
</tr>
</tbody></table>
<br><br>
</td>
</tr>
<tr bgcolor="#eeeeee">
<td height="40" valign="middle" width="15">&nbsp;</td>
<td valign="middle" width="374">
<font face="arial" size="-1">
<b>
<font color="red">Important:</font></b> Please change your password on next
Logon.</font><br>
</td>
<td align="center" valign="middle" width="57">
&nbsp;
</td>
</tr>
</tbody></table>
</td>
</tr>
</tbody></table>
<br>
<font color="#000000" face="ms sans serif" size="-2">
<center><font color="#000000" face="ms sans serif" size="1">Copyright © 2005 <a
href="http://www.quickmail.com/" target="TOP">QuickMail.com</a>
```

```
India Limited.
All Rights Reserved.
</center></font>
</center>
</form>
</body></html>
```

10.13 addressbookaddnew.jsp

```
/*
*       This file is used to add new
*       contact in the address book
*       It allows quick add as well
*       as full add.
*/



<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<%! String userid ; %>
<%
  userid  =   (String)session.getAttribute("userid");
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Welcome to Quickmail:</TITLE>
<META http-equiv=Content-Type content="text/html; charset=windows-1252">

<META content="Microsoft FrontPage 4.0" name=GENERATOR></HEAD>
<BODY text=#494949 vLink=#0066ff aLink=#0066ff link=#0066ff bgColor=#ffffff
leftMargin=0 topMargin=0 marginheight="0" marginwidth="0"><!-- Start Top Bar --
>
<TABLE cellSpacing=0 cellPadding=0 width="782" border=0>
  <TBODY>
  <tr>
    <TD align=left width="631">
    <IMG
    alt="QuickMail" src="../images/LogoBig.jpg" border=0 width="143"
height="34"></TD>
  </tr>
  <tr>
    <TD bgcolor="#336699" height="7" width="631">
    <IMG height=7 border=0 width="7"></TD>
  </tr>
  <tr>
    <TD vAlign=top width="631">
    <a href="../images/clogin.jpg"><IMG src="../images/clogin.jpg" border=0
    valign="TOP" align="left"></a></TD>
  </tr>
  <tr>
    <TD bgColor="#336699" height=5 width="631">
      <TABLE cellSpacing=0 cellPadding=0 border=0 bgcolor="#336699">
        <TBODY>
        <TR>
          <TD height=1 bgcolor="#336699"></TD></TR></TBODY></TABLE></TD>
  </tr>
  </TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <tr>
    <TD class=sb1 width="100%" bgColor=#F2F0D0
      height=24>&nbsp;&nbsp;&nbsp;&nbsp;<B>Welcome <%=userid%>,</B><BR>
      <SCRIPT>
if (x==0)
{
document.write("<TABLE cellSpacing=0 WIDTH=417 cellPadding=0
border=0><TR><TD height=1></TD></TR></TABLE>");
}
if (x==1)
```

```
{
document.write("<TABLE cellSpacing=0 WIDTH=630 cellPadding=0
border=0><TR><TD height=1></TD></TR></TABLE>");
}
</SCRIPT>
  </TD>
  <TD width=1 bgColor=#F2F0D0>
   <TABLE cellSpacing=0 cellPadding=0 border=0>
    <TBODY>
    <TR>
     <TD width=1 height=1></TD></TR></TBODY></TABLE></TD>
  <TD class=sb2 align=middle width=118 bgColor=#F2F0D0>
   <TABLE cellSpacing=0 cellPadding=0 width=118 border=0>
    <TBODY>
    <TR>
     <TD width=1 height=1></TD></TR></TBODY></TABLE></TD>
  <TD width=1 bgColor=#F2F0D0>
   <TABLE cellSpacing=0 cellPadding=0 border=0>
    <TBODY>
    <TR>
     <TD width=1 height=1></TD></TR></TBODY></TABLE></TD>
  <TD class=sb2 align=middle width=136 bgColor="#F2F0D0">&nbsp;</TD>
 </tr>
 <tr>
  <TD bgColor=#336699 colSpan=5 height=1>
   <TABLE cellSpacing=0 cellPadding=0 border=0>
    <TBODY>
    <TR>
     <TD height=1></TD></TR></TBODY></TABLE></TD>
 </tr>
 </TBODY></TABLE><!-- End Top Bar --><!-- Main Table Starts -->
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
```

```
  <TBODY>
  <TR>
    <TD vAlign=top width=168 ><!-- Left Navi starts -->
<TABLE cellSpacing=0 cellPadding=0 width=168 border=0 bgColor=#336699>
        <TBODY>
        <TR>
          <TD class=sb2 bgColor=#ecf9ff height=24>&nbsp; &nbsp; <A
            href="inbox.jsp">Inbox</A>
        </TD></TR>
        <TR>
          <TD></TD></TR>
        <TR>
          <TD class=sb2 bgColor=#ecf9ff height=24>&nbsp; &nbsp; <A
            href="compose.jsp">Write
            Mail</A> </TD></TR>
        <TR>
          <TD></TD></TR>
        <TR>
          <TD class=sb2 bgColor=#ffffff height=24>&nbsp; &nbsp; <B>Address
            Book</B> </TD></TR>
        <TR>
          <TD></TD></TR>
        <TR>
          <TD class=sb2 bgColor=#ecf9ff height=24>&nbsp; &nbsp; <A
            href="settings">My Settings</A>
          </TD></TR>
        <TR>
          <TD></TD></TR>
        <TR>
          <TD class=sb1 bgColor=#ecf9ff height=24>&nbsp; &nbsp; • <A
            href="logout.jsp">Logout</A>
        </TD></TR>
        <TR>
          <TD></TD></TR>
        <TR>
          <TD><IMG height=3 hspace=0 src="../images/f002.gif" width=168
            border=0><BR></TD></TR>
        <TR>
          <TD><IMG height=3 hspace=0 src="../images/f002.gif" width=168
            border=0><BR></TD></TR>
        <TR>
          <TD><IMG height=3 hspace=0 src="../images/f002.gif" width=168
            border=0><BR></TD></TR>
        <TR>
          <TD>
          </TD></TR></TBODY></TABLE>

<!-- Left Navi Ends --></TD>
    <TD width=1 bgColor=#67c1e9>
      <TABLE cellSpacing=0 cellPadding=0 border=0>
        <TBODY>
        <TR>
          <TD width=1 height=1></TD></TR></TBODY></TABLE></TD>
    <TD vAlign=top width="100%">
      <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
        <TBODY>
        <TR>
          <TD width=312 height=42><FONT class=sb4>&nbsp;&nbsp;<font
color="#355995">Address
            Book</font></FONT> </TD></TR>

<SCRIPT language=JavaScript>
 function MyCancel()
 {
        location.href="inbox.jsp";
}

function CheckAll(chk)
{
        for (var i=0;i < document.forms[0].elements.length;i++)
        {
                var e = document.forms[0].elements[i];
                if (e.type == "checkbox")
                {
                        e.checked = chk.checked;
                }
        }
        populate_compose()
}

function CheckGroupAll(chk)
{
        for (var i=0;i < document.forms[0].elements.length;i++)
        {
                var e = document.forms[0].elements[i];
                if (e.type == "checkbox")
                {
                        e.checked = chk.checked;
                }
        }
        populate_grp_compose()
}


function deleteAddr()
{
        document.forms[0].FormName.value="checkeddelete";
        document.forms[0].submit();
}



function quickAdd()
{
        document.forms[1].FormName.value="add_nick";
        document.forms[1].submit();
}

</SCRIPT>

        <FORM
        action="/html/_servlet_"  method=post><INPUT type=hidden value=edit_nick
name=FormName> <INPUT
        type=hidden name=ord> <INPUT type=hidden name=lowchar> <INPUT
        type=hidden name=srt> <INPUT type=hidden name=start><INPUT
        type=hidden name=x value="addressbookaddnew">
        <INPUT type=hidden name=userid value=<%=userid%>>
        <TR>
          <TD class=sb2 colSpan=2>&nbsp;&nbsp;Individual
            Contacts &nbsp;&nbsp; </TD></TR></TBODY></TABLE>
      <TABLE cellSpacing=0 cellPadding=0 border=0>
        <TBODY>
        <TR>
          <TD width=1 height=12></TD></TR></TBODY></TABLE>
      <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
        <TBODY>
        <TR bgColor=#d2d2d2>
```

```
     <TD class=sb2 colSpan=2 height=24 bgcolor="#07409D">&nbsp;<font
color="#FFFFFF">&nbsp;&nbsp;&nbsp;<B>Add
      the following address then click on Quick Add »</B>
      </font>
   </TD></TR></TBODY></TABLE>
   <TABLE cellSpacing=0 cellPadding=0 border=0>
    <TBODY>
    <TR>
     <TD width=1 height=1></TD></TR></TBODY></TABLE>
   <TABLE cellSpacing=0 cellPadding=0 width="100%" bgColor=#eeeeee
    border=0><TBODY>
    <TR>
     <TD class=sb2 height=24 bgcolor="#07409D">&nbsp;&nbsp;<font
color="#FFFFFF">&nbsp;&nbsp;Quick Add »
      </font> <FONT
      color=#ff0000>(* required fields)</FONT>
</TD></TR></TBODY></TABLE>
   <TABLE cellSpacing=0 cellPadding=0 border=0>
    <TBODY>
    <TR>
     <TD width=1 height=1></TD></TR></TBODY></TABLE>
   <TABLE cellSpacing=0 cellPadding=0 width="100%" bgColor=#ecf9ff
    border=0><TBODY>
    <TR>
     <TD colSpan=4 height=5 bgcolor="#FFFFFF">
      <TABLE cellSpacing=0 cellPadding=0 border=0>
       <TBODY>
       <TR>
        <TD width=1 height=5></TD></TR></TBODY></TABLE></TD></TR>
    <TR>
     <TD class=sb2 align=right width=86 bgcolor="#FFFFFF"><FONT
color=#355995><B>Nickname :
      </B></FONT></TD>
     <TD class=sb1 width=140 bgcolor="#FFFFFF"><INPUT type=hidden size=12
value=bca_ignou
      name=orgnick> &nbsp;<INPUT size=14  name=nick> <FONT
      class=sb5 color=#ff0000>*</FONT></TD>
     <TD class=sb2 align=right width=98 bgcolor="#FFFFFF"><FONT
color=#355995><B>Email :
      </B></FONT></TD>
     <TD class=sb1 height=35 bgcolor="#FFFFFF">

      &nbsp;<INPUT size=25  name=email>
      <FONT class=sb5 color=#ff0000>*</FONT> </TD></TR>
    <TR>
     <TD class=sb2 align=right width=86 bgcolor="#FFFFFF"><FONT
color=#355995><B>First Name
      : </B></FONT></TD>
     <TD class=sb1 width=150 bgcolor="#FFFFFF">&nbsp;<INPUT size=14
name=fname> </TD>
     <TD class=sb2 align=right width=98 bgcolor="#FFFFFF"><FONT
color=#355995><B>Last Name
      : </B></FONT></TD>
     <TD class=sb1 height=35 bgcolor="#FFFFFF">&nbsp;<INPUT size=14
name=lname> </TD></TR>
    <TR>
     <TD class=sb2 align=right bgcolor="#FFFFFF"><IMG height=10 hspace=5
      src="asasas_files/f004.gif" width=5 border=0><FONT
      color=#355995><B>Mobile : </B></FONT></TD>
     <TD class=sb1 bgcolor="#FFFFFF">&nbsp;<INPUT size=14 name=cell>
</TD>
     <TD class=sb2 align=right bgcolor="#FFFFFF"></TD>
     <TD class=sb1 height=35 bgcolor="#FFFFFF"></TD></TR>
    <TR>
```

```
          <TD class=sb2 align=right bgcolor="#BBC7E3">&nbsp; </TD>
          <TD colSpan=3 bgcolor="#BBC7E3">
            <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
              <TBODY>
              <TR>
                <TD><INPUT class=ab_sbttns title="Quick Edit" type=submit
value="Quick Add">
                </TD>
                <TD class=sb1>&nbsp;&nbsp;<FONT color=#355995>Click Quick Add
                  button to Save the Address or continue to fill the information
                  below.</FONT> </TD></TR></TBODY></TABLE></TD></TR>
        <TR>
          <TD colSpan=4 bgcolor="#FFFFFF">
            <TABLE cellSpacing=0 cellPadding=0 border=0>
              <TBODY>
              <TR>
                <TD
height=10></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE>
      <TABLE cellSpacing=0 cellPadding=0 border=0>
        <TBODY>
        <TR>
          <TD height=1></TD></TR></TBODY></TABLE>
      <TABLE cellSpacing=0 cellPadding=0 width="100%" bgColor=#eeeeee
        border=0><TBODY>
        <TR>
          <TD class=sb2 height=24 bgcolor="#07409D">&nbsp;<font
color="#FFFFFF">&nbsp;&nbsp;&nbsp;Contact Numbers »
            (optional)</font> </TD></TR></TBODY></TABLE>
      <TABLE cellSpacing=0 cellPadding=0 border=0>
        <TBODY>
        <TR>
          <TD width=1 height=1></TD></TR></TBODY></TABLE>
```

```
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
 <TBODY>
 <TR>
  <TD colSpan=4 height=5>
   <TABLE cellSpacing=0 cellPadding=0 border=0>
    <TBODY>
    <TR>
     <TD width=1 height=5></TD></TR></TBODY></TABLE></TD></TR>
 <TR>
  <TD class=sb2 align=right width=86><FONT color=#355995><B>Residence
   : </B></FONT></TD>
  <TD class=sb1 width=140>&nbsp;<INPUT size=14 name=homeph> </TD>
  <TD class=sb2 align=right width=98><FONT color=#355995><B>Work :
   </B></FONT></TD>
  <TD class=sb1 height=35>

   &nbsp;<INPUT size=14 name=workph> </TD></TR>
 <TR>
  <TD class=sb2 align=right width=86><FONT color=#355995><B>Fax :
   </B></FONT></TD>
  <TD class=sb1 width=150>&nbsp;<INPUT size=14 name=fax> </TD>
  <TD class=sb2 align=right width=98>&nbsp; </TD>
  <TD class=sb1 height=35>&nbsp; </TD></TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=0 border=0>
 <TBODY>
 <TR>
  <TD height=1></TD></TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=0 width="100%" bgColor=#eeeeee
 border=0><TBODY>
 <TR>
  <TD class=sb2 height=24 bgcolor="#07409D">&nbsp;&nbsp;<font
color="#FFFFFF">&nbsp;&nbsp;Address » (optional)</font>
```

```
      </TD></TR></TBODY></TABLE>
    <TABLE cellSpacing=0 cellPadding=0 border=0>
     <TBODY>
     <TR>
      <TD width=1 height=1></TD></TR></TBODY></TABLE>
    <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
     <TBODY>
     <TR>
      <TD colSpan=4 height=5>
       <TABLE cellSpacing=0 cellPadding=0 border=0>
        <TBODY>
        <TR>
         <TD width=1 height=5></TD></TR></TBODY></TABLE></TD></TR>
     <TR>
      <TD class=sb2 align=right width=86><FONT color=#355995><B>Address :
       </B></FONT></TD>
      <TD class=sb1 colSpan=3 height=35>&nbsp;<INPUT size=50 name=address>
      </TD></TR>
     <TR>
      <TD class=sb2 align=right width=86><FONT color=#355995><B>City :
       </B></FONT></TD>
      <TD class=sb1 width=140>&nbsp;<INPUT size=14 name=city> </TD>
      <TD class=sb2 align=right width=98><FONT color=#355995><B>State :
       </B></FONT></TD>
      <TD class=sb1 height=35>
       <SCRIPT>
   &nbsp;<INPUT size=14 name=state> </TD></TR>
     <TR>
      <TD class=sb2 align=right><FONT color=#355995><B>Postal code :
       </B></FONT></TD>
      <TD class=sb1>&nbsp;<INPUT size=14 name=pin> </TD>
      <TD class=sb2 align=right><FONT color=#355995><B>Country :
          </B></FONT></TD>
         <TD class=sb1 height=35>&nbsp;<INPUT size=14 name=country>
        </TD></TR></TBODY></TABLE>
      <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
        <TBODY>
        <TR>
          <TD height=5>
          </TD></TR>
        </TBODY></TABLE>
      <TABLE cellSpacing=0 cellPadding=0 width="100%" bgColor=#d2d2d2
        border=0><TBODY>
        <TR>
          <TD align=right width=200 height=32
bgcolor="#BBC7E3">&nbsp;&nbsp;<INPUT class=ab_sbttns1 title="Edit Contact"
type=submit value=Add>
          </TD>
          <TD align=right width=130 bgcolor="#BBC7E3"><INPUT class=ab_sbttns1
title=Cancel onclick=MyCancel(); type=button value=Cancel>
          </TD>
          <TD bgcolor="#BBC7E3">

          </TD></TR></TBODY></TABLE>

    </TD></TR></TBODY></TABLE><!-- Main Table Ends --><!-- Footer starts -->
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <TR>
    <TD width="100%" bgColor=#336699 height=4>
      <TABLE cellSpacing=0 cellPadding=0 border=0>
        <TBODY>
        <TR>
```

```
          <TD width="100%"
height=4></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE>
<br><br>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TR>
    <TD align=left colSpan=2 height=1 bgcolor="#336699">
    <IMG height=1
      src="../images/pobtrans.gif" width="100%"></TD></TR>
  <TR>
    <TD vAlign=top align=left><font face="Arial" size="1">Copyright © 2005 <a
href="http://www.quickmail.com/" target="TOP">QuickMail</a> India Limited.
All Rights Reserved. </font></TD>
    <TD vAlign=top align=right><FONT face=Arial size=-2>
    <a href="../contact_us.html">Contact Us</a> •
    <a href="../quickmail_terms_conditions.htm">Terms and Conditions </a>&nbsp;•
    <a href="../quickmail_privacy_policy.htm">Privacy
      Policy</a></FONT></TD></TR>
  <TR>
    <TD vAlign=top align=right colSpan=2><FONT face=Arial size=1>This page is
      managed by MySql</FONT></TD></TR></TABLE>
<p><BR>
</FORM></p>
</BODY></HTML>
```

10.14 deletemail.jsp

```
/*
*       This file is used to delete
*       the mails from the inbox.
*/



<%@ page import="java.util.*,javax.activation.*, javax.mail.*, javax.mail.internet.*
" %>
<%! String userid, userpass,dd,mm,dat,tim,other ; %>

<%
  userid  =   (String)session.getAttribute("userid");
  userpass= (String)session.getAttribute("userpass");
%>



<%
String tot=request.getParameter("num");

int mailnumber=-1;
try
{
  mailnumber=Integer.parseInt(request.getParameter("mailnum"));
}
catch(Exception e)
{
  System.out.println("Nan....................................");
}
```

```
int mailcount=0,num=-1,k=-1;
int  mailnum[]={-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,};
String temp;
boolean selAll=false;
StringTokenizer st=null;
if(tot != null)
{
  st = new StringTokenizer(tot,",");
  temp=st.nextToken();
  while(st.hasMoreTokens())
  {
    temp=st.nextToken();
    try
    {
      if(temp.equals("Select All"))
        selAll=true;
      else
      {
        mailnum[mailcount]=Integer.parseInt(temp);
        mailcount++;
      }
    }
    catch(Exception e)
    {
      System.out.println("Nan error :::: "+e.getMessage());
    }
  }
}
if(selAll==true)
{
  mailcount=mailnumber;
  System.out.println("select all mail for delete");
```

```
}

Store s1=null;
Folder inbox=null;
try
{
  other=request.getParameter("sendmessagedelete");
  if(other==null)
    other="other";
  Properties props=new Properties();
  props.put("mail.store.protocol","pop3");

  props.put("mail.pop3.host",quickmail.user.ip.IP.ip);
  Session s=Session.getInstance(props,null);

  s1=s.getStore("pop3");

  s1.connect(quickmail.user.ip.IP.ip,userid+quickmail.user.ip.IP.mailatrate,userpass);

  inbox =s1.getFolder("INBOX");
  inbox.open(Folder.READ_WRITE);
  Message[] m=inbox.getMessages();
  int del=-1;
  for(int j=0; j <=mailcount  ; j++)
  {
    del=mailnum[j];
    if(del != -1)
    {
      m[del].setFlag(Flags.Flag.DELETED,true);
    }
     out.println("ffffffffffffffffffff "+del );
  }
```

```
out.println("wwwwwwwwwww");
  inbox.close(true);
}
catch(Exception e)
{
System.out.println("Error in deleting mail "+e.getMessage());
}
finally
{
  s1.close();
}
%>
<!--"javascript:window.close()"-->
<jsp:forward  page= "inbox.jsp"/>
```

10.15 filehandle.jsp

```
/*
*         This file is used in sending attachments
*         with the mail. It is still
*         experimental.
*/

<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<%
  quickmail.user.login.attach.FileTest ftest= new
quickmail.user.login.attach.FileTest();
  String s=request.getParameter("filenm");
  String txt=request.getParameter("list");
  String str="";
  if(s==null)
  {
     s="1";
     response.sendRedirect("filehandleerror.jsp?x=filehandleerror&val=null");
  }
  else
   str=ftest.fileCheck(s);

 if(!str.equals("ok"))
   response.sendRedirect("filehandleerror.jsp?x=filehandleerror&val=null");
%>

<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>Attach your file</title>
</head>
<script language="javascript">
        function addinlist()
        {
                if(document.frm.attach.value != "")
                {
                        document.frm.list.value= document.frm.list.value +"\n" +
document.frm.attach.value;
    var href="/html/_servlet_?x=filehandle&filenm="+document.frm.attach.value;
    location.href=href;
                }
                else
                {
                        alert("No File is to Attach ");
                }
        }
        function done1()
        {
   document.parentNode.nodeName=list;
   document.activeElement.insertAdjacentText="dddddddddddddddddd"
   document.frm.submit();
        }

</script>
<body>
<TABLE cellSpacing=0 cellPadding=0 width="350"
background="../images/pobtrans.gif" border=0 dwcopytype="CopyTableRow">
 <TR>
```

```
    <TD align=left width="348">
    <IMG
      alt="QuickMail" src="../images/LogoBig.jpg" border=0></TD></TR>
  <TR>
    <TD bgcolor="#336699" height="7" width="348">
    <IMG height=7 border=0 width="7"></TD></TR>
  <TR>
    <TD vAlign=top width="348">
    <a href="../images/clogin.jpg"><IMG src="../images/clogin.jpg" border=0
    valign="TOP" align="left"></a></TD></TR></TABLE>
<form name="frm" method="POST" action="fdf">
   <table border="0" width="52%">
    <tr>
      <td width="100%" colspan="2"><font size="5" color="#8AB1D9">&nbsp;File
        Attach With Mail</font></td>
    </tr>
    <tr>
      <td width="100%" colspan="2">Brows Files&nbsp; <input type="File"
name=attach  ></td>
    </tr>
    <tr>
      <td width="15%">&nbsp;</td>
      <td width="85%"><input type="button" value="     Attach File into List     "
name="fileattach" onclick="addinlist()" ></td>
    </tr>
    <tr>
      <td width="100%" colspan="2">&nbsp;<TEXTAREA readonly name=list
rows=5 wrap=physical cols=39 size="58" value=<%
if(txt==null && str.equals("ok")==true)
  out.println(s);
      %>>
</TEXTAREA> </td>
```

```
    </tr>
    <tr>
      <td width="1%">&nbsp; </td>
      <td width="114%"><input onclick="done1()" type="button" value="
Done                 " name="done"> </td>
    </tr>
    <tr>
      <td width="100%" colspan="2">&nbsp; </td>
    </tr>
  </table>
</form>

&nbsp;
<TABLE cellSpacing=0 cellPadding=0 width="350" border=0>
  <TBODY>
  <TR>
    <TD align=left colSpan=2 height=1 bgcolor="#336699" width="427">
    <IMG height=1
      src="../images/pobtrans.gif" width="100%"></TD></TR>
  <TR>
    <TD vAlign=top align=left width="400"><font face="Arial" size="1">Copyright
© 2005 <a href="http://www.mymail.com/" target="TOP">MyMail.com</a> India
Limited.
All Rights Reserved. </font></TD>
    <TD vAlign=top align=right width="25"></TD></TR>
  <TR>
    <TD vAlign=top align=right colSpan=2 width="427"><FONT face=Arial
size=1>This page is managed Apache</FONT></TD></TR></TBODY></TABLE>

</body>
</html>
```

10.16 registrationerror.jsp

```
/*
*        This jsp is used to display the
*        error in the registration process.
*/


<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<html>
<HEAD>
<TITLE>Login Error</TITLE>
<style type="text/css">
</style>

</HEAD>
<BODY BGCOLOR=#FFFFFF LINK=#0066FF ALINK=#0066FF
VLINK=#0066FF leftmargin=0 marginwidth=0 marginheight=0 topmargin=0>
<FORM>
<TABLE WIDTH=100% CELLSPACING=0 CELLPADDING=0 BORDER=0
height="499">
<TR>
<TD height="51">
<BR><img src="../images/LogoBig.jpg" border=0 hspace=15 vspace=4 width="106"
height="22"><br>
</TD>
</TR>
<TR>
<TD height=6 bgcolor=#336699>
<TABLE cellSpacing=0 cellPadding=0 border=0><TR><TD
height=6></TD></TR></TABLE>
```

```
</TD>
</TR>
<TR>
<TD bgcolor=#eeeedd align=center height="407">
<BR><BR><BR>
<table width=406 border=0 cellpadding=1 cellspacing=0 bgcolor=#ffffff>
<tr>
<td>
<table width=404 border=0 cellpadding=0 cellspacing=0 bgcolor=#FFE1E1
style="border-collapse: collapse" bordercolor="#111111">
<tr>
<td align=center>
<TABLE cellSpacing=0 cellPadding=0 border=0><TR><TD
height=9></TD></TR></TABLE>
<img src="../images/e002.gif" border=0 width="43" height="44"><br><BR>
<font class=sb4 color="#FF0000">Sorry.</font><BR><BR>
<font class=sb3>
Your registration failed.<BR>
<TABLE cellSpacing=0 cellPadding=0 border=0><TR><TD
height=9></TD></TR></TABLE>
Username already registered.<TABLE cellSpacing=0 cellPadding=0
border=0><TR><TD height=9></TD></TR></TABLE>
Please select another Username and then try again.<BR>
<TABLE cellSpacing=0 cellPadding=0 border=0><TR><TD
height=9></TD></TR></TABLE>
Try something unique.<BR><BR>
<a href="javascript:history.go(-1)"><B>Click here to go back</B></a><BR><BR>
</font>
</td>
</tr>
</table>
</td>
```

```
</tr>
</table>
<BR><BR><BR><BR><BR>
</TD>
</TR>
<TR>
<TD class=sb1 align=center height=35>
<font face="Arial" size="1">Copyright © 2005 <a href="login.html"
target="TOP">QuickMail.com</a> India Limited.
All Rights Reserved.</font>
</TD>
</TR>
</TABLE>
</FORM>
</BODY>
</html>
```

10.17 logout.jsp

```
/*
*	This jsp is used in the logout process
*	it invalidates the user session.
*/


<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<html>
<HEAD>
<TITLE>Logout successfully</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF LINK=#0066FF ALINK=#0066FF
VLINK=#0066FF leftmargin=0 marginwidth=0 marginheight=0 topmargin=0>
<FORM>
<TABLE WIDTH=100% CELLSPACING=0 CELLPADDING=0 BORDER=0
height="499">
<TR>
<TD height="51">
<BR><img src=../images/LogoBig.jpg border=0 hspace=15 vspace=4 width="106"
height="22"><br>
</TD>
</TR>
<TR>
<TD height=6 bgcolor=#336699>
<TABLE cellSpacing=0 cellPadding=0 border=0><TR><TD
height=6></TD></TR></TABLE>
</TD>
</TR>
```

```
<TR>
<TD bgcolor=#eeeedd align=center height="407">
<BR><BR><BR>
<table width=406 border=0 cellpadding=1 cellspacing=0 bgcolor=#ffffff>
<tr>
<td>
<table width=404 border=0 cellpadding=0 cellspacing=0 bgcolor=#FFE1E1
style="border-collapse: collapse" bordercolor="#111111">
<tr>
<td align=center>
<TABLE cellSpacing=0 cellPadding=0 border=0><TR><TD
height=9></TD></TR></TABLE>
<img src=../images/e002.gif border=0 width="43" height="44"><br><BR>
<font class="sb4" color="#FF0000">LOGOUT</font><BR><BR>
<font class=sb3>
Your logout successfully.<BR>
<TABLE cellSpacing=0 cellPadding=0 border=0><TR><TD
height=9></TD></TR></TABLE>
<p>
<a href="../login.html"><b>Login again on QuickMail&nbsp;</b></a><BR><BR>
</font>
</td>
</tr>
</table>
</td>
</tr>
</table>
<BR><BR><BR><BR><BR>
</TD>
</TR>
<TR>
<TD class=sb1 align=center height=35>
```

```
<font face="Arial" size="1">Copyright © 2005 <a
href="http://www.quickmail.com/" target="TOP">QuickMail.com</a> India Limited.
All Rights Reserved.</font>
</TD>
</TR>
</TABLE>
</FORM>
</BODY>
</html>
```

10.18 attachprocess.jsp

```
/*
*        This file is used in sending attachments
*        with the mail. It is still
*        experimental.
*/


<%@ page contentType="text/html;charset=WINDOWS-1252"%>
<%@ page import="java.util.*" %>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=WINDOWS-1252">
<META NAME="GENERATOR" CONTENT="Oracle JDeveloper">
<TITLE>
Hello World
</TITLE>
</HEAD>
<BODY>
<%
  String str=request.getParameter("list");
  System.out.println(str);
  StringTokenizer st= new   StringTokenizer(str,",");
String s="";
String collect="";
  while(st.hasMoreTokens())
  {
    s=st.nextToken();
    if(quickmail.user.login.attach.FileTest.fileCheck(s).equals("ok"))
    {
      out.println("file ok "+s);
      collect+=s;
    }
    else
      out.println("file errorooooooooooooooo"+s);
  System.out.println(s);
  }
  response.sendRedirect("compose.jsp?list="+collect);

%>
</BODY>
</HTML>
```

DATABASE SCRIPT

11.1 DATABASE SCRIPT

```
/* Database: quickmail                         */

/* Creation Date 18/3/2005                     */

/* DATABASE CREATION SCRIPT                    */


/* START OF SCRIPT                             */


CREATE DATABASE quickmail;

USE quickmail;


CREATE TABLE login (

      userid VARCHAR(20) NOT NULL PRIMARY KEY,

      upassword VARCHAR(15) NOT NULL,

      mpassword VARCHAR(15) NOT NULL);




CREATE TABLE user (

      userid VARCHAR(20) NOT NULL PRIMARY KEY,

      fname VARCHAR(15) NOT NULL,

      lname VARCHAR(15) NOT NULL,

      sex   CHAR(1) NOT NULL,

      mstatus CHAR(1) NOT NULL,

      country VARCHAR(20) NOT NULL,

      state VARCHAR(20) NOT NULL,

      city VARCHAR(20) NOT NULL,

      street VARCHAR(20) NOT NULL,
      pin VARCHAR(8) NOT NULL,
      education VARCHAR(30) NOT NULL);
CREATE TABLE pwdrecall (
      userid VARCHAR(20) NOT NULL PRIMARY KEY,
      question VARCHAR(30) NOT NULL,
      answer VARCHAR(30) NOT NULL
      dob VARCHAR(10) NOT NULL);

CREATE TABLE addressbook (
      userid VARCHAR(20) NOT NULL PRIMARY KEY,
      nic_name VARCHAR(15) NOT NULL,
      email VARCHAR(40) NOT NULL,
      fname VARCHAR(15),
      lname VARCHAR(15),
      country VARCHAR(20),
      state VARCHAR(20),
      city VARCHAR(20),
      street VARCHAR(20),
      pin VARCHAR(8),
      mobile VARCHAR(20),
      resi VARCHAR(20),
      office VARCHAR(20));

/* END OF SCRIPT                                  */
```

CODE EFFICIENCY & OPTIMIZATION

12.1 CODE EFFICIENCY

I have tried to make the code as efficient as possible. I have used the MVC architecture. So a lot of work is done by the beans. A single bean is used for all type of database connectivity. So it allows any module of the whole program to connect to the database. Also bean allows code reusability as they can be used in other programs. Beans also help in portability for example this program is made for MySQL but other databases can also be used easily.

Also the use of servlet greatly increases the efficiency of the program. Servlet are compiled once and once initialized and run they handle all the requests. One instance can handle many requests, so there is no need to run multiple instances; this greatly reduces the load on server.

So by using the reusability of beans greatly helped in coding, as it reduced the effort to code and test the redundant part of the program again and again.

12.2 CODE OPTIMIZATION

I have used functions for redundant tasks; this greatly helps in code optimization as these functions can be called whenever needed. Also these functions are often written in java beans and then these java beans can be used by other pages, so reducing a lot of coding. This helps in reducing complexity. Also I have checked a lot of values entered by users using java scripts before sending them to the server; this greatly helps in reducing the load of the server, as the initial checking is done by the user machine. This increases performance of the server.

DATA INTEGRITY & VALIDATION CHECKS

13.1 DATA INTEGRITY

Data integrity is very important in any project because invalid data is of no use so various measures are taken for maintaining data integrity. In web based e-mail the most important data for the smooth functioning of the system is the data contained in the login and pwdrecall table, so various measures are taken to maintain data integrity in these two tables. There are two steps for maintaining data integrity:

- ❖ **Value Constraints and Ranges**
- ❖ **Validation Checks**

13.2 VALUE CONSTRAINTS AND RANGES

The first step for maintaining data integrity is to identify various value ranges of various attributes. This is done in the design phase. In the web-based email system user can make changes to the data regarding to him in all the four tables so for the smooth functioning of system few constraints are defined on the data entered by the user which are as follows:

Attribute	**Value constraint and range**
userid	userid can contain a-z and 0-9 and must be maximum of 20 characters.
upassword	Can contain any character but minimum six.
mpassword	Only administrator can enter it.
question	One question from a set of questions maximum of 30

	characters.
answer	Not important for program. Can contain any character. But not blank.
dob	Must be in date format MM/DD/YYYY
fname	Not important for program. Can contain any character.
lname	Not important for program. Can contain any character.
sex	Can contain only M or F.
mstatus	Some status.
country	Must be some country.
city	Not important for program. Can contain any character.
street	Not important for program. Can contain any character.
pin	Not important for program. Can contain any character.
education	Not important for program. Can contain any character.
nic_name	Not important for program. Can contain any character.
email	An email address.
mobile	Not important for program. Can contain any character.
resi	Not important for program. Can contain any character.
office	Not important for program. Can contain any character.

13.3 VALIDATION CHECKS

Validation checks are the second step in maintaining data integrity. These are implemented in the coding phase of the project development. When the user enters data validation checks are performed over it before using it. If it is found to violate its value range then an appropriate message is shown to the user.

The project developed also performs a lot of validations on the data entered. Some of them are given below:

Attribute	**Validation Checks Performed**
userid	Is thoroughly checked for value constraints before checking it with the server.
upassword	Checked only at registration time to be more than six characters.
mpassword	Only administrator can enter it so not checked.
question	Drop down list.
answer	Checked for blank entry.
dob	Drop down list for entering the value.
fname	Checked for blank entry.
lname	Checked for blank entry.
sex	Drop down list for entering the value.
mstatus	Drop down list for entering the value.
country	Can contain any character but maximum of 15 characters.
state	Not important not checked.
city	Not important not checked.
street	Not important not checked.
pin	Not important not checked.
education	Not important not checked.
nic_name	Not important not checked.
email	Mostly automatically entered.
mobile	Not important not checked.
resi	Not important not checked.
office	Not important not checked.

TESTING

14.1 TESTING

It is the job of programmer to test, as far as possible, that all parts of the programs work correctly. It should be realized that complete testing is not possible except in the case of the most trivial program. One can never be completely certain that all errors have been removed, but sufficient test can be performed to give a reasonable measure of confidences in the program. A good testing requires the following:

- ❖ A through knowledge and understanding of what the system is supposed to do.
- ❖ Plan out in advance what ought to be tested.
- ❖ To work out expected results for each of the test cases.
- ❖ Writing out test plan.

14.2 TYPES OF TESTING

The processes of testing are the steps to validate and prepare a system for final implementation. The programmer must do the following testing:

- ❖ **Unit Testing** – In unit testing the programmer tests all the individual modules that are making the system. Unit testing gives stress on the modules independently of each other. This helps in easily detecting the logical errors in the modules.

  In my project all the individual files that were coded were tested individually for any errors. They were tested for the inputs they take and were tested to see what happens when we enter an illegal value for a field. After the results of an individual module were found satisfactory it was then integrated with others. Test cases with reports are given in 14.3.

- **Integration Testing –** After testing all the individual modules all the modules are integrated and then tested for errors. The errors resulting due to interactions of modules are found in this testing.
  In my project after the individual modules were integrated they were again tested for errors. If some error was detected then the interface between the modules was corrected. This testing is not as lengthy as unit testing.

- **System Testing** – It is one of the most important and essential part of the system development phase, after designing and developing the system. In system testing all the subsystems are gathered into one pool and then the whole system is tested to determine whether it meets the user requirements.
  In my project after I had incorporated the system with all the other components system testing was done to test for any errors that might have remained. Testing was done to test the efficient working of all the subcomponents together.

14.3 TESTING CASES AND REPORTS

LEGEND

A lot of symbols are used in the testing data and reports which are given as follows:

1.	N/A	**:**	Not Applied.
2.	A	**:**	Apply
3.	S	**:**	Select this field.
4.	BFP	**:**	Blank Field Prohibited.
5.	BFA	**:**	Blank Field can be Applied.
6.	MBE	**:**	Must Be Existing.
7.	EMA	**:**	Error Message Appeared.

14.3.1 UNIT TESTING RESULT

Tester Name **:** Amir Ansari

Programmer Name **:**Amir Ansari

UNIT TESTING OVERALL RESULT			
S. No.	**Module Description**	**Result**	**Successful/Failed**
1	Sign In	As Expected	Successful
2	Sign Up	As Expected	Successful
3	Compose	As Expected	Successful
4	Address Book	As Expected	Successful
5	Forget Password	As Expected	Successful

UNIT Test Case Status:	***Successful***

14.3.2 UNIT TEST CASES AND RESULTING TESTING DATA

The test cases used in unit testing along with the resulting data according to modules is given below:

1 SIGN IN

Tester Name	:Amir Ansari
Programmer Name	:Amir Ansari
Module Name	: Sign In

ARRAY OF VALUES								
Field Name	**Blank Entry**	**Numeric Entry**	**Alphabet Entry**	**@ Entry**	**Special Symbol**	**Expected Result**	**Observed Result**	**Test Result**
User ID	BFP	A	A	N/A	N/A	EMA	EMA	**Pass**
User Password	BFP	A	A	N/A	N/A	EMA	EMA	**Pass**

Module Test Status:	***Successful***

2 SIGN UP

Tester Name	:Amir Ansari
Programmer Name	:Amir Ansari
Module Name	: Sign Up (Registration)

ARRAY OF VALUES

Field Name	Blank Entry	Numeric Entry	Alphabet Entry	@ Entry	Special Symbol	Expected Result	Observed Result	Test Result
Userid	BFP	A	A	N/A	N/A	EMA	EMA	**Pass**
User password	BFP	A	A	N/A	N/A	EMA	EMA	**Pass**
Password Retype	BFP	A	A	N/A	N/A	EMA	EMA	**Pass**
Secret Question	BFP	A	A	A	A	EMA	EMA	**Pass**
Secret Answer	BFP	A	A	A	A	EMA	EMA	**Pass**
Day	BFP	A	N/A	N/A	N/A	EMA	EMA	**Pass**
Month	BFP	N/A	A	N/A	N/A	EMA	EMA	**Pass**
Year	BFP	A	N/A	N/A	N/A	EMA	EMA	**Pass**
Sex	BFP	N/A	S	N/A	N/A	EMA	EMA	**Pass**
Marital Status	BFP	N/A	S	N/A	N/A	EMA	EMA	**Pass**
Country	BFP	N/A	A	N/A	N/A	EMA	EMA	**Pass**
City	BFP	N/A	A	N/A	N/A	EMA	EMA	**Pass**
Pin	BFP	A	N/A	N/A	N/A	EMA	EMA	**Pass**
Education	BFA	A	N/A	N/A	N/A	EMA	EMA	**Pass**
First name	BFP	N/A	A	N/A	N/A	EMA	EMA	**Pass**
Last Name	BFP	N/A	A	N/A	N/A	EMA	EMA	**Pass**

Module Test Status:	***Successful***

3 COMPOSE

Tester Name : Amir Ansari

Programmer Name : Amir Ansari

Module Name : Compose

ARRAY OF VALUES								
Field Name	**Blank Entry**	**Numeric Entry**	**Alphabet Entry**	**@ Entry**	**Special Symbol**	**Expected Result**	**Observed Result**	**Test Result**
To	BFP	A	A	MBF	N/A	EMA	EMA	Pass
Cc	BFA	A	A	A	N/A	EMA	EMA	Pass
Bcc	BFA	A	A	A	N/A	EMA	EMA	Pass
Subject	BFA	A	A	A	A	EMA	EMA	Pass

Module Test Status:	***Successful***

4 ADDRESS BOOK

Tester Name : Amir Ansari

Programmer Name :Amir Ansari

Module Name : Address Book

ARRAY OF VALUES								
Field Name	**Blank Entry**	**Numeric Entry**	**Alphabet Entry**	**@ Entry**	**Special Symbol**	**Expected Result**	**Observed Result**	**Test Result**
Nick Name	BFP	A	A	N/A	N/A	EMA	EMA	Pass
E-mail ID	BFP	A	A	MBE	N/A	EMA	EMA	Pass
First name	BFA	A	N/A	N/A	N/A	EMA	EMA	Pass
Last Name	BFA	A	N/A	N/A	N/A	EMA	EMA	Pass
Mobile	BFA	N/A	A	N/A	N/A	EMA	EMA	Pass
Residence	BFA	N/A	A	N/A	N/A	EMA	EMA	Pass
Work	BFA	N/A	A	N/A	N/A	EMA	EMA	Pass
Fax	BFA	N/A	A	N/A	N/A	EMA	EMA	Pass
Address	BFA	A	A	N/A	N/A	EMA	EMA	Pass
City	BFA	A	N/A	N/A	N/A	EMA	EMA	Pass
State	BFA	A	N/A	N/A	N/A	EMA	EMA	Pass
Country	BFA	A	N/A	N/A	N/A	EMA	EMA	Pass

Module Test Status:	***Successful***

5 FORGOT PASSWORD

Tester Name	: Fareedul Hai
Programmer Name	: Fareedul Hai
Module Name	: Compose

ARRAY OF VALUES								
Field Name	**Blank Entry**	**Numeric Entry**	**Alphabet Entry**	**@ Entry**	**Special Symbol**	**Expected Result**	**Observed Result**	**Test Result**
Username	BFP	A	A	N/A	N/A	EMA	EMA	Pass
Day	BFP	A	N/A	N/A	N/A	EMA	EMA	Pass
Month	BFP	N/A	A	N/A	N/A	EMA	EMA	Pass
Year	BFP	A	N/A	N/A	N/A	EMA	EMA	Pass
City	BFP	N/A	A	N/A	N/A	EMA	EMA	Pass
Country	BFP	N/A	A	N/A	N/A	EMA	EMA	Pass
Hint Question	BFP	A	A	N/A	N/A	EMA	EMA	Pass
Hint Answer	BFP	A	A	N/A	N/A	EMA	EMA	Pass

Module Test Status:	***Successful***

14.3.3 SYSTEM TESTS AND RESULTING TESTING DATA

System tests were done when the whole system was integrated as a whole. All the functions were then performed to test the proper working of the whole system. The test result data is given as follows:

Test Cases				
Case No.	**Step Description**	**Expected Result**	**Actual Result (if different from expected)**	**Success/ Failed**
1	Login using invalid data.	Access Denied		Success
2	Login using valid data.	Access Granted to Inbox.		Success
3	Message header selected and mouse clicked.	Message data shown.		Success
4	Delete pressed.	Message deleted.		Success
5	All message selected and delete pressed.	All selected messages deleted.		Success
6	Compose link clicked.	Compose page opens with all the address of address book shown.		Success
7	Send clicked without entering any address.	Error message shown.		Success
8	A contact clicked.	Address automatically filled in “To:” field.		Success
9	Send clicked after entering an address.	Message send and information shown and address automatically entered in contacts.		Success
10	Sign Up clicked.	Registration page shown.		Success
11	Registration page not completely filled and OK clicked.	Error Message shown.		Success
12	Registration page completely filled and OK clicked.	Registration successful message shown.		Success
13	Address book clicked.	Address book with all		Success

		contacts shown.		
14	New contact clicked.	New form shown to enter the details.		Success
15	Email address not filled and contact saved.	Error message shown.		Success
16	Email address and nick name filled and contact saved.	Contact successfully saved.		Success
17	Logout clicked.	Login again page shown.		Success
			System Test Status	***Successful***

After doing all the tests the working of project was found to be satisfactory. The program was found to be reliable and working in perfect condition.

SECURITY IMPLEMENTATION

15.1 SECURITY MECHANISM

This project as it deals with users personal data needs good security mechanism which is taken care of. Various measures have to be taken to take care of security in the whole project. Care of security is taken for individual servers as well as the whole system. Steps are taken to counter three types of security issues which are as follows:

- ❖ **Unauthorized access to database server.**
- ❖ **Unauthorized access to mail server.**
- ❖ **Steps taken against hacking of system.**

Some of the steps taken to counter the above problems are described below:

- The E-mail system is based on **JSP Model 2** advanced architecture, also known as **Model-View-Controller** or **MVC** architecture. In this architecture the control flow (or application flow) is embodied in another module called **controller.** As the entire user requests are handled by the controller only allowed commands can be used by the users. So a potential hacker can not get all the listing of files in the email website etc.

- The database can be configured to only allow queries generated by the programs of the local machine to run so remote user is unable to run a query.
- First of all at the time of registration the user is required to set a password which should be greater than six characters for better security.

- User password is independent of mail server password so user will not be able to access mail server directly in anyway through telnet etc.

- Passwords wherever entered are not shown instead asterisk (*) character is shown.

- When a user logins in the E-mail system his login details are not saved in a cookie on the local system for increased security as some other person can read the cookie details. The system maintains login **session details** on the server which it manages in **a hash table** till the user is logged in. As the user does logoff his details are erased from the server hash table.

- On pages that change user password or other setting the user is asked to enter the password again for better security.

- ❖ In case the user forgets his/her password and uses the password recall option to get the password the system resets the user password using a random password instead of a default password for increased security.

COST & EFFORT ESTIMATION OF PROJECT

16.1 COST ESTIMATION

Cost estimation is an important aspect of every project. For commercial projects this is one of the most important estimation. My project was only for educational purpose, so it didn’t needed much cost estimation. Cost of any software depends mainly upon three things which are as follows:

- ❖ **Hardware Needed**
- ❖ **Software Tools Needed**
- ❖ **Human Resource Needed**

This project as it uses the basic hardware doesn’t need any special hardware. All the hardware needed for this project is mostly present in every computer. A computer was freely available which was used for the development of this project. So there was no investment in hardware.

In case of software the JDeveloper IDE, all the other server software and tools needed for this project are freely available on the internet, so only a very small amount of money was needed to download them. So we can say that the investment on software was also close to nothing.

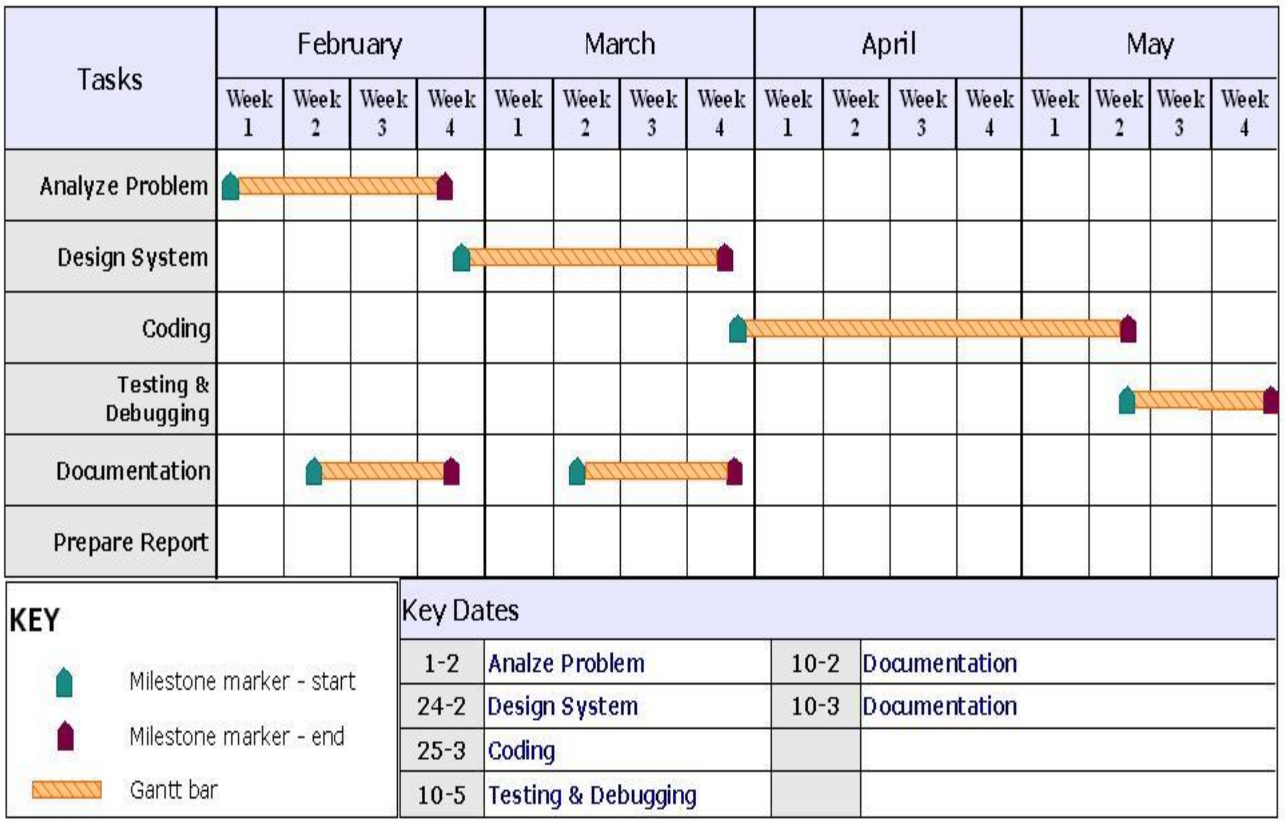
This software as it was developed by me alone and I did spend about a period of about seven months. So this was the biggest investment on this project. Also this was a new type of project for me as I haven't worked on J2EE platform before so it took a little more time than any experienced programmer might have taken.

After deciding all the three aspects the cost of this project can be expected to about **four person-months of an experienced programmer**.

PERT CHART & GANTT CHART

17.1

GANTT CHART FOR VERSION 1 - 4 MONTH TIME LINE



Key Dates

1-2	Analze Problem	10-2	Documentation
24-2	Design System	10-3	Documentation
25-3	Coding		
10-5	Testing & Debugging		

FUTURE SCOPE AND FURTHER ENHANCEMENT

18.1 FUTURE SCOPE

The E-mail system has a really good scope as E-mail is one of the most used services provided by computer networks. Every medium to large business organization have a large amount of correspondence between the employees, and as e-mail provides fast and cheapest correspondence, it is the most efficient way of correspondence so every organization needs it.

This E-mail system as it is based on **free open source** software which does not require any **licensing issues** can provide the cheapest and robust system. It requires only one time investment and unlike other commercial systems does not require buying licenses with every little increase of users. So business organizations will like using it.

Also a lot of components developed for this project can be reused for future projects with a little change. Few components that can be reused are password checking bean, password reminder etc. So we can say this project as a whole as well its components have a good future.

18.2 FURTHER ENHANCEMENTS

Although the present system is satisfying, many of the requirements of a web-based e-mail system. Yet there are few limitations of this system which can be easily deal with. The E-mail system currently supports text based emails and attachments of all types but with a little effort it will be able to send multimedia emails incorporating pictures and other objects. Also I will try to incorporate **voice mail** in this system so that it will be of great help for the users.

Currently the system as it is for an organization-website which provides email facility to the company employees so it requires the administrator to approve an email account before the new user can use his account but this can be easily changed so that the user does not need an approval from the administrator, after that this E-mail system can to be used as free for all or paid E-mail website.

Also a few cosmetic changes can be done to make the interface more beautiful. This will further enhance the user experience.

GLOSSARY

20 GLOSSARY

A

Attachment	An audio, video or other data file that is attached to an email message.

B

BCC	Blind Carbon Copy – Email Address entered in this field receives a copy of the email but does not know the other people addresses to which a copy of email is send.
Beans	Java uses the Bean's specification to allow the creation of software component that can be used by other developer and designers to build specialist applications.

Browser	Browser is client software that allows the user to display and interact with a hypertext document.

C

CC	Carbon Copy – Email Address entered in this field receives a copy of the email but also receives the address of all the people to which a CC of email is send.

D

DNS	Domain Name Server (or system) – An Internet service that translates domain names into IP addresses.
Download	To transfer a copy of a file from an Internet server to one's own computer.
DSN	DSN is needed to setup an ODBC Open Database Name (DSN) to be able to connect the database.

E

Email bounces	Email messages that fail to reach their intended destination. "Hard" bounces are caused by invalid email addresses, whereas "soft" bounces are due to temporary conditions, such as overloaded inboxes.

Email client	The software that recipients use to read email. Some email clients have better support for HTML email than others.
Email header	The section of an email message that contains the sender's and recipient's email addresses as well as the routing information.

F

Freeware	A free computer program usually made available on the Internet or through user groups.

H

Header	The first part of an email message that contains the controlling and meta-data such as the subject, origin and destination email addresses, the path an email takes.
Host name	The name of a computer on the Internet (such as www.abc.com).
HTML	Hyper Text Markup Language – The most commonly used coding language for creating Web pages.

HTTP	Hypertext Transfer Protocol is a protocol used on the Web to transit Hypertext documents.

I

Internet	The largest worldwide computer network.
Intranet	Contrary to the public Internet, an intranet is a private network inside a company or organization.
IP address	An IP address is a unique identifier for a computer on the Internet. It is written as four numbers separated by periods. Each number can range from 0 to 255. Before connecting to a computer over the Internet, a Domain Name Server translates the domain name into its corresponding IP address.

J

JDBC	Java uses a set of classes and interfaces called Java database connectivity to interact with database.
JSP	JavaServer Pages (JSP) provides Web Developer with a framework to create dynamic content on the server using HTML and XML, templates and Java code which is secure, fast and independent of server platform.

L

LAN	Local Area Network, which is a computer network, although geographically limited, usually to the same building, office, etc.

M

MIME	Multi-Purpose Internet Mail Extensions – An extension of the original Internet email standard that allows users to exchange text, audio or visual files.

O

Operating system	A program that manages all other programs in a computer, such as Windows or UNIX.

P

Plain text	It is referred to text in an email message that contains no formatting elements.
POP	Post Office Protocol – A protocol used to retrieve email from a mail server. Most email clients use either the POP or the newer IMAP protocol.

Protocol	The set of formal rules that describe how to transmit data, especially across a network of computers.

Q

Query	A subset of records in a database. These are normally used to select some specific data from the database.

S

Server	A program that acts as a central information source and provides services to programs in the same or other computers. The term can either refer to a particular piece of software, such as a WWW server, or to the machine on which the software is running.
SMTP	Simple Mail Transfer Protocol – A protocol used to send email on the Internet. SMTP is a set of rules regarding the interaction between a program sending email and a program receiving email.
Snail mail	Traditional or surface mail sent through postal services such as the Indian Postal Service.
Subject line	The part of an email message where senders can type what the email message is about. Subject lines are considered important by

	email marketers because they can often influence whether a recipient will open an email message.

T

TCP / IP	Transmission Control Protocol / Internet Protocol – This is the protocol that defines the Internet. TCP / IP were originally designed for the UNIX operating system, but are today available for every major kind of computer operating system.

U

URL	Uniform Resource Locator – The address of a file or Web page accessible on the Internet (for example, http://www.quickmail.com).
User Interface	A set of controls such as buttons, commands and other devices that allow a user to operate a computer program.

X

XML	Extensible Markup Language – A flexible way to create standard information formats and shares both the format and the data on the World Wide Web.

****