

MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING



CENG491 FALL 2006

SENIOR DESIGN PROJECT

DETAILED DESIGN REPORT

A Unified News Exchange Server with NNTP, Mail, Web and RSS

MULTIWAY

by

SANZATU YAZILIM

1. INTRODUCTION.....	4
1.1 Purpose of This Document.....	4
1.2 Problem Definition.....	4
1.3 Project Scope and Goals.....	6
1.4 Project Application Areas.....	7
1.5 Project Design Overview.....	7
1.6 Design Issues.....	9
2. DATA DESIGN.....	10
2.1 Data Design Overview.....	10
2.2 Database Entities and Relationships.....	11
2.3 Database Table SQL's	12
2.4 Data Dictionary.....	13
3. SYSTEM STRUCTURE.....	18
3.1 System Core.....	18
3.2 Web Access Module.....	19
3.3 Newsreader Access Module.....	22
3.4 Mail Transfer Module.....	24
3.5 RSS Module.....	26
3.6 System Management Module.....	27
4. SYSTEM DATA FLOW.....	28
5. OBJECT-ORIENTED CONCEPTS.....	32
5.1 System Classes.....	32
5.2 Interactions among Classes.....	35
6. SYNTAX SPECIFICATIONS.....	42
7. USER INTERFACE DESIGN.....	43
7.1 User Web Interface Design.....	43
7.2 Administrator Interface Design.....	58
8. SYSTEM REQUIREMENTS.....	63
8.1 Minimal Hardware Requirements.....	63

8.2 Minimal Software Requirements.....	63
9. PROJECT SCHEDULE.....	64
9.1 Project Task Set.....	64
9.2 Updated Gantt Chart.....	66
10. TESTING.....	69
10.1 Unit Testing.....	69
10.2 Integration Testing.....	70
10.3 Performance Tests.....	70

1. INTRODUCTION

1.1 Purpose of This Document

The purpose of this document is to explain the design specifications of the project which we have been thinking on for a long time and have made definite a couple of times ago. In this report, we intend to give detailed information about what are our solutions and how they fulfill the problem requirements. Thanks to our studies throughout the semester, we have developed our sight to the problem and to the solution. We will present the modular specifications, data design, detailed structure and components of the system and we will also explain those using ER and UML diagrams.

1.2 Problem Definition

In this century interaction between people all around the world is supplied by computers. We get to know more about life, communicate with other people, do most of our works via these computers now. Being a member of online forums and groups which connect people from all over the world, sending and receiving e-mails from our friends or worldwide working companies, connecting to other computers which are located in other sides of the world, reading numerous kinds of articles from websites, getting the news from online magazines or newspapers are just simple daily activities of our lives.

In this project are to build up a system which supports several protocols and methods that are used in order to achieve the communication between server and client computers. Some of these protocols, methods and their application areas are as follows.

The World Wide Web ("WWW" or simply the "Web") is a global, read-write information space. WWW is available via to the internet which is network of networks all over the world. On the World Wide Web, information resources are identified by unique, global identifiers called Uniform Resource Identifiers (URIs). For viewing a web page or other resources on the Web, the unique URI of it should be typed on a web browser. After finding the IP address of the web server which contains the resource, a Hypertext Transfer Protocol (HTTP) request is send to this web server and then requested

information is received by the client in quick succession. HTTP is a method used to transfer or convey information on the Web. HTTP, a request/response protocol between clients and servers, defines how messages are formatted and transmitted, and what actions Web servers and browsers (clients) should take in response to various commands.

NNTP, short for *Network News Transfer Protocol*, is the protocol used by both web browsers and servers to send, distribute and receive USENET postings and news articles. It is designed so that news articles are stored in a central database or file system allowing a subscriber to select only those items he/she wishes to read. A NNTP server is a computer that has the task to collect a copy of news messages from the discussion groups and to allow users to read and send messages to the same groups. Indexing, cross-referencing, and expiration of aged messages are also provided.

RSS (Rich Site Summary) is a method that is used for accessing headlines of changes in different internet pages. First the users subscribe to the internet pages they are interested in. By that way they can get the headlines of a change in one of those internet pages on the RSS reader interface and can see the content of it by clicking on that headline of the change. By the help of RSS, people access regularly changing web contents that are interested without losing time in the internet.

Simple Mail Transfer Protocol (SMTP), is Internet's standard host-to-host mail transport protocol and traditionally operates over TCP, port 25. In other words, a UNIX user can type `telnet hostname 25` and connect to an SMTP server, if one is present. After connecting to SMTP server, a dialog is passed between the client computer and the server for authentication of the client, then the message text is transferred to one or more recipients specified (in most cases these recipients are verified to exist). SMTP is a "push" protocol that does not allow one to "pull" messages from a remote server on demand. That is, one can only send emails but not receive emails using SMTP. To do this a mail client must use POP3 or IMAP.

In our project, we are to design a system which combines all of these methods under one communication system. Users will be able to interact with the system throughout web browsers, news readers and RSS readers. The system will also provide users with emails of new messages of selected groups by the users. For example, when a user posts an article to the newsgroups, the article will be fed to the RSS clients, seen on the forum page, and delivered to the users by email.

All these features will be implemented as separate modules and the communication between these modules and the data storage will be provided by the system core. Depending on the analysis phase we may implement security check mechanisms in this core. Also we may locate the core and the data storage in the same machine since core will usually be interacting with the data storage.

Connections between clients and servers will be implemented in an efficient and secure fashion in order to create a reliable system. We plan to implement a filtering feature which will prevent the newsgroups from invalid (spam containing) messages. We also want to use SSL (Secure Socket Layer) with HTTP and NNTP. Our system will also be platform-independent to able to be set up on machines using different operating systems. In addition, we think that a user-friendly interface is important for the www-based implementation part of the system.

1.3 Project Scope and Goals

Multiway is a server application software package by which a communication system including Hypertext Transfer Protocol (HTTP), Network News Transfer Protocol (NNTP), Simple Mail Transfer Protocol (SMTP), and RSS feed can be built up. This system will mainly provide the following facilities;

- ✓ Access to the system and read/write messages via web and newsreaders.
- ✓ Automatic email posting from system to the registered users of the system.
- ✓ System's registered users capable of posting articles to newsgroups via email.
- ✓ RSS feed support.

- ✓ Spam filtering for received messages and received emails.
- ✓ File transfer among members.

1.4 Project Application Areas

Our project product has wide application areas. All the applications needing interaction with subscribed users are possible to be implemented by Multiway. Some of these areas are below:

- ✓ **Distance or Traditional Education:** The interaction between teachers and students may be provided by the newsgroups. There may be separate groups for different courses which give the students the opportunity to discuss their ideas and by our chat system which we may implement as an additional feature students can talk to each other online. Students can also upload their homework to the area created by the chief of the newsgroup of the course.
- ✓ **Companies:** The interaction between managers and workers may be provided by the newsgroups. There may be separate groups for different departments which give the workers the opportunity to discuss the projects and exchange ideas. Workers can also upload their weekly reports to the area created by the chief of the newsgroup of the department.
- ✓ **Online Forums:** This application can use our web module. People exchange ideas and share files on the web by subscribing to the forums formed by different newsgroups. File sharing will be managed by the chief of each newsgroup.

1.5 Project Design Overview

We will supply a package program in an executive which will be installed by the customer to some machine. After the executive is run and package is installed, the machine will be used as the main server of the system and there will be HTTP, NNTP, SMTP servers installed in it. Those will be working interactively while the users make requests to the server machine.

In the system we supply, there are registered users and these users are checked by the system for authentication when logging in. And there are newsgroups which users can write and read articles, delete or update the articles they wrote before. There are main groups and their sub newsgroups. A newsgroup may be public or private. A public newsgroup can be read or written by all the registered users while only some of the registered users are allowed to read or write private newsgroups.

There is an administrator of the system which is responsible for the system management. Admin can make changes on the system using a program installed on main server machine. He/she creates the user accounts and newsgroups. While creating a newsgroup, he/she decides its permissions and the users allowed to read/write that newsgroup if it is private. He/she can select where to keep log files of the users and XML files of newsgroups for RSS functionality. He can also decide when the posts would expire and are sent to the archive.

There are four ways (WWW, newsreaders, email and RSS) to read the articles in a newsgroup to post articles to a newsgroup.

The first one of the ways by which a user can use the system is World Wide Web. A user writes HTTP URL of the system and makes request to connect to the server machine. After authentication is done, there are lots of functionalities the user can use via the user interface we designed. He/she can read/write/reply articles in the newsgroups he/she is allowed to. He/she can search a specific article and save some of the articles in his/her account. He/she can select the option of reading XML files of the news via his/her RSS Reader, and also subscribe to the mailing lists of the groups. For each private newsgroup, there are downloadable files and uploaded files. It is the newsgroup chief's responsibility to put downloadable files on the web and also to spare the area for the users to upload files.

The second way of accessing to the system is via newsreaders. The user can set his/her newsreader to access the system and make use of all newsreader functionalities (read/write/follow-up/search articles) on the system.

As mentioned before, via the web, a user can be registered to the mailing lists of some newsgroups. After that registration, the system will send articles to the user's registered email address as mail and the user will be able to read articles he/she is interested in

without directly accessing the system. Also, he/she can send articles to the system and reply coming articles via his/her email address registered in the system. This also makes the user easily send articles without a direct access to the system.

The last way of accessing to the system is via RSS Readers. A user can see which newsgroups provide RSS functionality via the web and can read articles on those newsgroups and make comments on the articles via his/her RSS Reader. This also provides user's access to the recent news easily.

All of the sent articles via different ways are kept in the same data storage part and for example a user accessing the system via the web can see a post sent via email at the same time it is sent. That's, all those ways mentioned are interacted to each other.

After explaining user side of the system, let us talk about how these functionalities are done. There are open source servers used in the system and a core provides the interaction and communication between them. We use INN (InterNetNews) as NNTP Server, Apache as HTTP Server and Postfix as SMTP Server. We use INN's file system to save all the messages from all different sources and we make INN work all the time inside to serve the messages to the users for reading and save the sent messages in its file system. We chose this way not to ease the communication of different system modules and not to duplicate messages. In fact, this provides the interaction between different access ways of Multiway.

This was an overview of the system and throughout our report, we will explain all those modules in detail.

1.6 Design Issues

✓ Modularity and Interaction

Our system's main feature is that it can be accessed via different ways which form the system modules. So, modularity is the most important constraint. All the modules must perform their tasks completely and also they must properly interact to each other. That's, one change applied to the system via one of the access ways must be reflected to the other access method's results immediately.

✓ **Language**

We implement a number of protocols and their servers in our system. They are supposed to communicate each other to perform system tasks. We use open source servers for these tasks of the system core. Each of the servers we use is written in different programming languages. In order to integrate them into one big core, we intend to use Java as the main language, because it has packages for server implementation and for providing their communication. We may also use PHP and Perl for the web design of our system.

✓ **User Interface**

We make a system of news exchange between people and it will directly be used through some user interface. So, GUI is an important aspect of our project. There are different user interfaces for different user types and so easy use is one of the most important issues of the design of our system. We took this into consideration while creating user interfaces.

2. DATA DESIGN

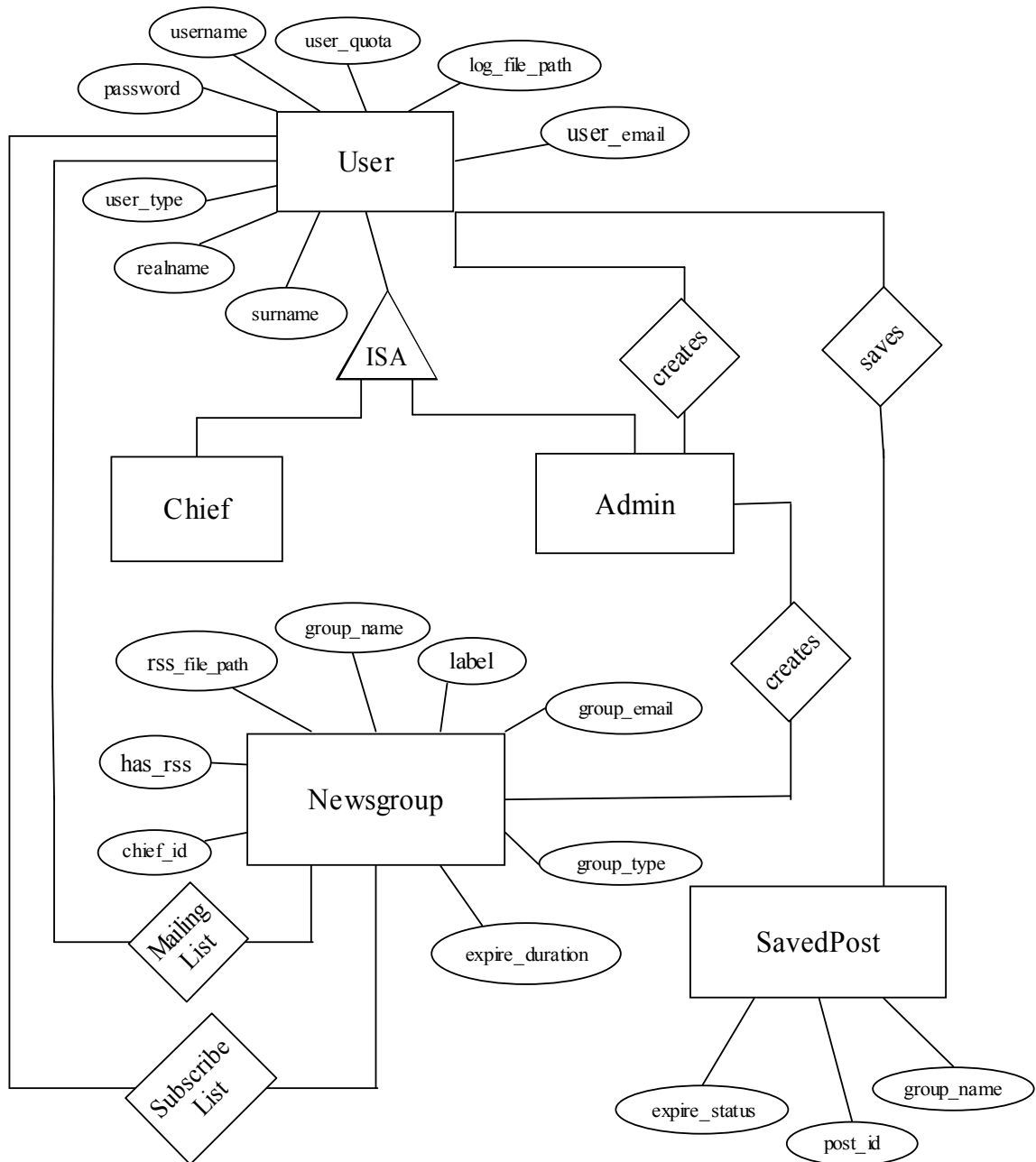
2.1 Data Design Overview

Our system uses NNTP open source server INN's file system to save all the messages from all different sources. INN stores articles in individual text files in a directory structure. For example, an article *employer* in newsgroup *company.humanresources* is stored as *var/spool/news/articles/company/humanresources/employer* relative to the root. Individual article data object has attributes as in the ER diagram.

In addition to file system, we also use database for keeping some required data. Those include information about the users, newsgroups and their relationships. We also keep information about saved posts in the database in order to decide where to find and access them when the user requests to see his/her account.

Below in the next section is the structure of the system database presented in ER diagrams. The MySQL commands creating those tables are in Section 2.3 and lastly, the description of objects and their attributes are presented in the Section 2.4.

2.2 Database Entities and Relationships



2.3 Database Table SQL's

```
CREATE TABLE User(username VARCHAR(20),
                  password VARCHAR(20),
                  user_type INTEGER,
                  realname VARCHAR(30),
                  surname VARCHAR(30),
                  user_email VARCHAR(100),
                  log_file_path VARCHAR(100),
                  user_quota INTEGER,
                  PRIMARY KEY (username));

CREATE TABLE Newsgroup(group_name VARCHAR(100),
                       label VARCHAR(100),
                       group_email VARCHAR(200),
                       group_status VARCHAR(10),
                       expire_duration INTEGER,
                       chief_id INTEGER,
                       has_rss INTEGER,
                       rss_file_path VARCHAR(100),
                       PRIMARY KEY (group_name));

CREATE TABLE MailingList(group_name VARCHAR(100),
                          username VARCHAR(20),
                          PRIMARY KEY (group_name, username),
                          FOREIGN KEY (group_name)
                              REFERENCES Newsgroup,
                          FOREIGN KEY (username)
                              REFERENCES User);

CREATE TABLE SubscribeList(group_name VARCHAR(100),
                            username VARCHAR(20),
                            PRIMARY KEY (group_name, username),
                            FOREIGN KEY (group_name)
                                REFERENCES Newsgroup,
                            FOREIGN KEY (username)
                                REFERENCES User);

CREATE TABLE SavedPost(username VARCHAR(20),
                        group_name VARCHAR(100),
                        post_id INTEGER,
                        expire_status INTEGER,
                        PRIMARY KEY (username, post_id, group_name)
                        FOREIGN KEY (username)
                            REFERENCES User,
                        FOREIGN KEY (group_name)
                            REFERENCES Newsgroup);
```

2.4 Data Dictionary

Name : User
Alias : -
Where/How used : When information of the users is needed.
Description : Every actor using the system is defined to be a user. That includes the administrator and newsgroup chiefs.

Name : username
Alias : -
Where/How used : While logging into the system.
Description : Username is a unique name given to the users for the system security. The users will enter their username together with their passwords to log into system.

Name : password
Alias : -
Where/How used : While logging into system.
Description : The password is to secure the system. Unauthorized users cannot log into the system. The user's password is converted to MD5 hash and checked with the one in the database.

Name : user_type
Alias : -
Where/How used : While logging into system.
Description : If 0, means regular user; if 1, means administrator; and if 2, means newsgroup chief. All the public and private newsgroups can be accessed by admin and newsgroup chiefs.

Name : realname
Alias : -
Where/How used : When messages in the newsgroups are shown.
Description : When a user is added to the system, his/her name is also specified by the administrator.

Name : surname
Alias : -
Where/How used : When messages in the newsgroups are shown.
Description : When a user is added to the system, his/her name is also specified by the administrator.

Name : user_email
Alias : -
Where/How used : When the user sends mail to newsgroups and when the system sends mail to the user.
Description : When a user is added to the system, his/her email address is also specified and the emails coming from that address are kept in the system and also the user can be added to the mailing list of the newsgroups if he/she wants.

Name : log_file_path
Alias : -
Where/How used : When user activities are to be known by the user/system.
Description : All the processes done by the user is recorded in the file system under that specified path.

Name : user_quota
Alias : -
Where/How used : When the user keeps the messages that he/she wants in his/her account.

Description : User quota specifies the size of user's account. The user can save posts in his/her account until the entire quota is used.

Name : Newsgroup

Alias : -

Where/How used : When information about the newsgroups is needed.

Description : -

Name : group_name

Alias : -

Where/How used : When the groups are shown on user interfaces.

Description : Specifies the name of the general group the newsgroup is connected to.

Name : label

Alias : -

Where/How used : When the groups are shown on user interfaces.

Description : Specifies the name of the newsgroup and unique for every newsgroup. Email address of the subgroups will be formed by concatenating the group name and the label.

Name : group_email

Alias : -

Where/How used : When the user sends email to newsgroups.

Description : Formed by concatenating the group name and the label of the newsgroup.

Name : group_type
Alias : -
Where/How used : When the newsgroups the user can read/write are shown on the user's interface.
Description : Each newsgroup is public or private to users. If a newsgroup is public, all the registered users can read/write it, while if it is private, only some of the registered users can read/write it. When a user logs in to the system, the newsgroups the user can read/write are specified according to the group_type and if it's private, according to username.

Name : expire_duration
Alias : -
Where/How used : When deciding on whether some of the posts are still shown on the interfaces or old and to be deleted from the interfaces (sent to the archive).
Description : Specified by the admin when the newsgroup is created. It is a period of time indicated by days beginning on the day the post is sent. Expired messages can be searched by the user but are not seen on the regular user interface.

Name : chief_id
Alias : -
Where/How used : When file operations of the newsgroup are done.
Description : When admin creates a private newsgroup, he/she will assign it a chief that has a unique id.

Name : has_rss
Alias : -
Where/How used : When rss option is shown on the user interface.

Description : If 1, then the newsgroup has RSS feed, and if it is 0, it has not. That value is specified by the admin when the newsgroup is created.

Name : rss_file_path

Alias : -

Where/How used : When XML file of a newsgroup is shown to the user.

Description : If the attribute 'has_rss' is 1, then the newsgroup has RSS feed, and its XML file is kept under 'rss_file_path'.

Name : MailingList

Alias : -

Where/How used : When the system sends messages coming to a newsgroup to users.

Description : Each newsgroup has a mailing list. Users can add themselves to this list using web interface if they want and they can read messages via their email addresses.

Name : SubscribeList

Alias : -

Where/How used : When the system has to decide which users can see which newsgroups on their interface.

Description : Each newsgroup has a subscribed users list. This list is specified when a newsgroup is created and then may be edited by the administrator.

Name : SavedPost

Alias : -

Where/How used : When the user saves a post or wants to access the posts he/she saved before.

Description	: When the user chooses to save a post he/she interested in, its newsgroup and post_id is kept in this database table.
Name	: expire_status
Alias	: -
Where/How used	: When the user wants to read his/her saved posts.
Description	: If 0, that means the post has not expired yet and it can be still seen on the user interface. If 1, that means the post has expired and was sent to the archive. According to its value, file system search area changes when the saved post is shown to the user.

3. SYSTEM STRUCTURE

Our system is mainly composed of a threaded and secure message exchange core and modules providing different access methods to this core. There are basic core functionalities which are reading, writing, updating and canceling a message. Different modules of the system provide those functionalities to be done via different ways. Those modules are Web Access Module, Mail Transfer Module, Newsreader Access Module and RSS Module. There is also a fifth one, System Management Module, which provides the management functionality for the administrator.

Below are the detailed explanations of our core and the extension modules.

3.1 System Core

In the core of our system, there is our NNTP server INN and there is the program part which provides the communication of different modules with the core. There are also three layers of the core where some basic functionalities are embedded. Those layers are Authentication, Logfile Keeping and Mail Alerting, outer to inner. To access the core, the Authentication layer must firstly be passed by checking user info from the database. Then, a request of the core functionalities (read, write, update, delete a message) actuates the program part mentioned above. If the request is a read request, the desired message is

brought from INN's file system to the related access module and if it is a write request, the composed message is put in INN's file system from the related access module.

Logfile Keeping layer of system core continuously updates users' log files under the admin-specified directories. In a log file, a specific user's login history is kept for a period of time.

Mail Alerting Layer alerts Mail Transfer Module whenever a post is sent to the system from different sources in order for that module to send that message to users as email.

3.2 Web Access Module

This module mainly provides access to the core using HTTP via a web browser. After writing URL of the system to the web browser, the user is first authenticated by HTTPS. After he/she logs in to the system, the allowed newsgroups and their articles are sent to web server from INN's file system.

When the user makes a read request, the desired article is brought from INN's file system by the script mentioned above in the first section and it is displayed in the web browser of the user.

When the user writes a message and makes a send request, the posted article will be converted to an NNTP message. Then the converted message will be sent to INN's file system by the script mentioned above in the first section.

A user may register himself to mailing list of a newsgroup using the web interface. When this registration is done by some user, this user and registered newsgroup is saved in 'MailingList' table of the database.

A user may save an article in his/her account via the web which means that the article is saved in 'savedPost' table of database. He/she can also see the articles in his/her account via the web. It is also possible for the user to change his personal information via the web. This will affect the corresponding 'User' table.

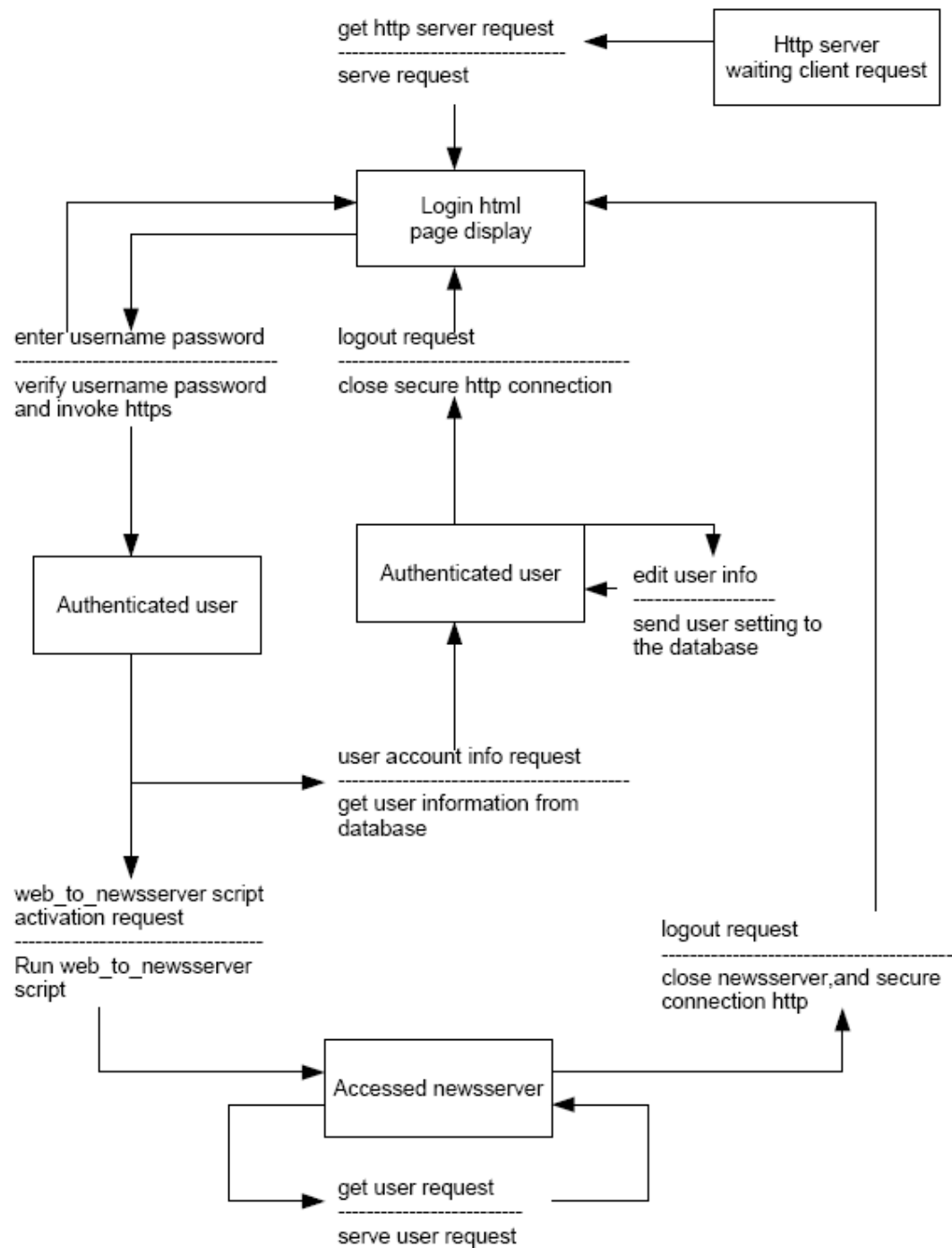
Web access module also provides the user to search the articles according to the subject. It is ensured that the searching will only be in the newsgroups that the user has the permission to access. For the searching task, a file search script will be developed. This program will search INN's file system and also the archive. Related articles are

brought from INN's file system by the script mentioned above in the first section and will be shown on the web browser after searching is complete.

A user may upload files to or download files from a newsgroup. An uploaded file will be saved in the file directory of that newsgroup. And a file to be downloaded is brought from the file directory of that newsgroup. It is ensured that the user can only upload files to and download files from newsgroups that he/she has the permission to access.

Using the web interface, a user can see whether or not a newsgroup provides RSS functionality.

➤ State Transition Diagram for Web Access Module:



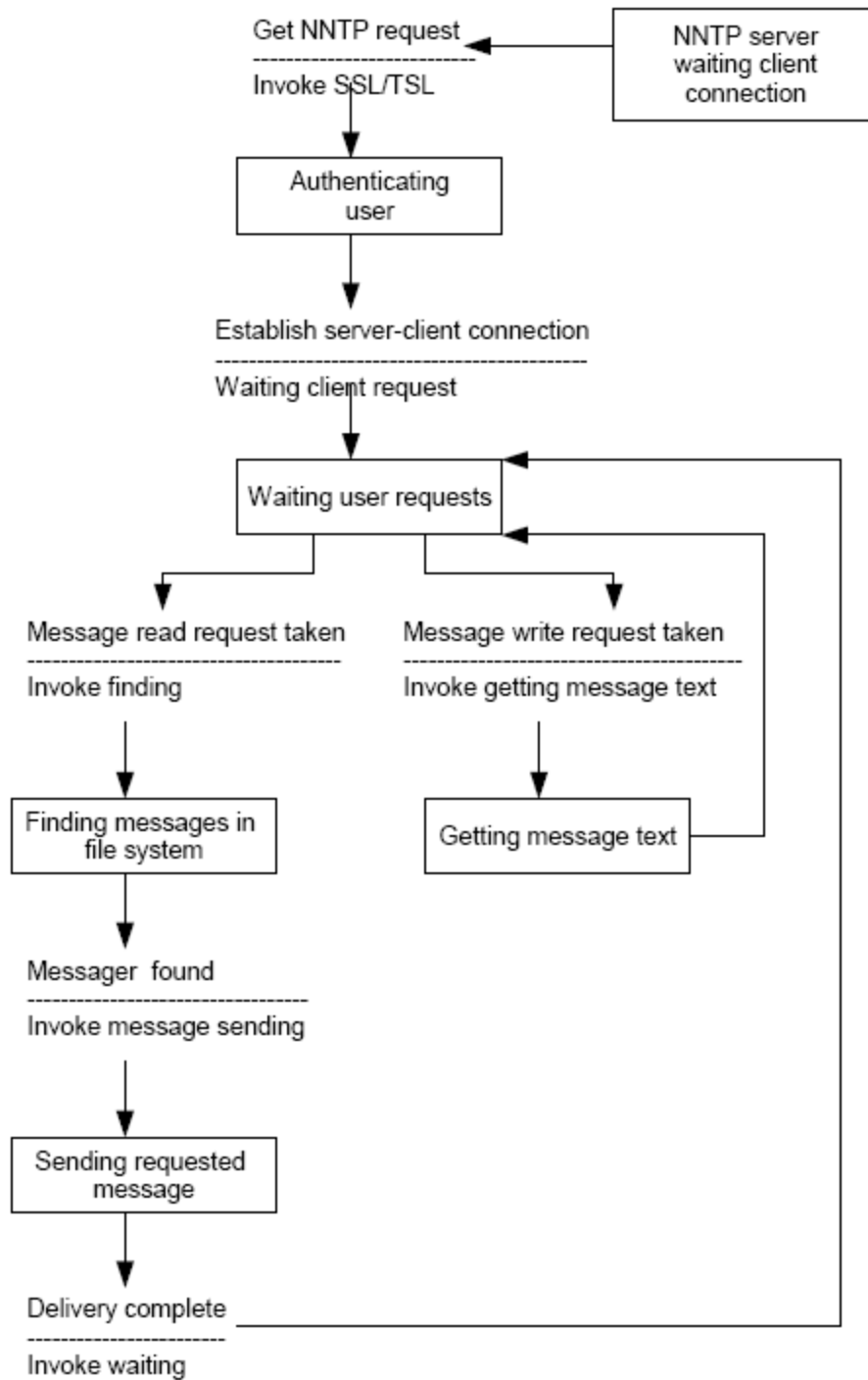
3.3 Newsreader Access Module

This module mainly provides access to the core using newsreaders.

Because newsreaders use NNTP for access to newsgroups, there is no need for message type conversion for newsreader access module. However, authentication is still needed. The authenticity of the users will be provided by SSL via the news server. Once the user has been authenticated, only the newsgroups that user has read/write permission are seen by the user on his/her newsreader.

The posted messages will be sent to NNTP Server INN automatically since newsreaders use NNTP for access to newsgroups, and the NNTP Server INN is on our core. On a read request, the desired articles are again automatically brought to the user's newsreader.

➤ State Transition Diagram for Newsreader Access Module:



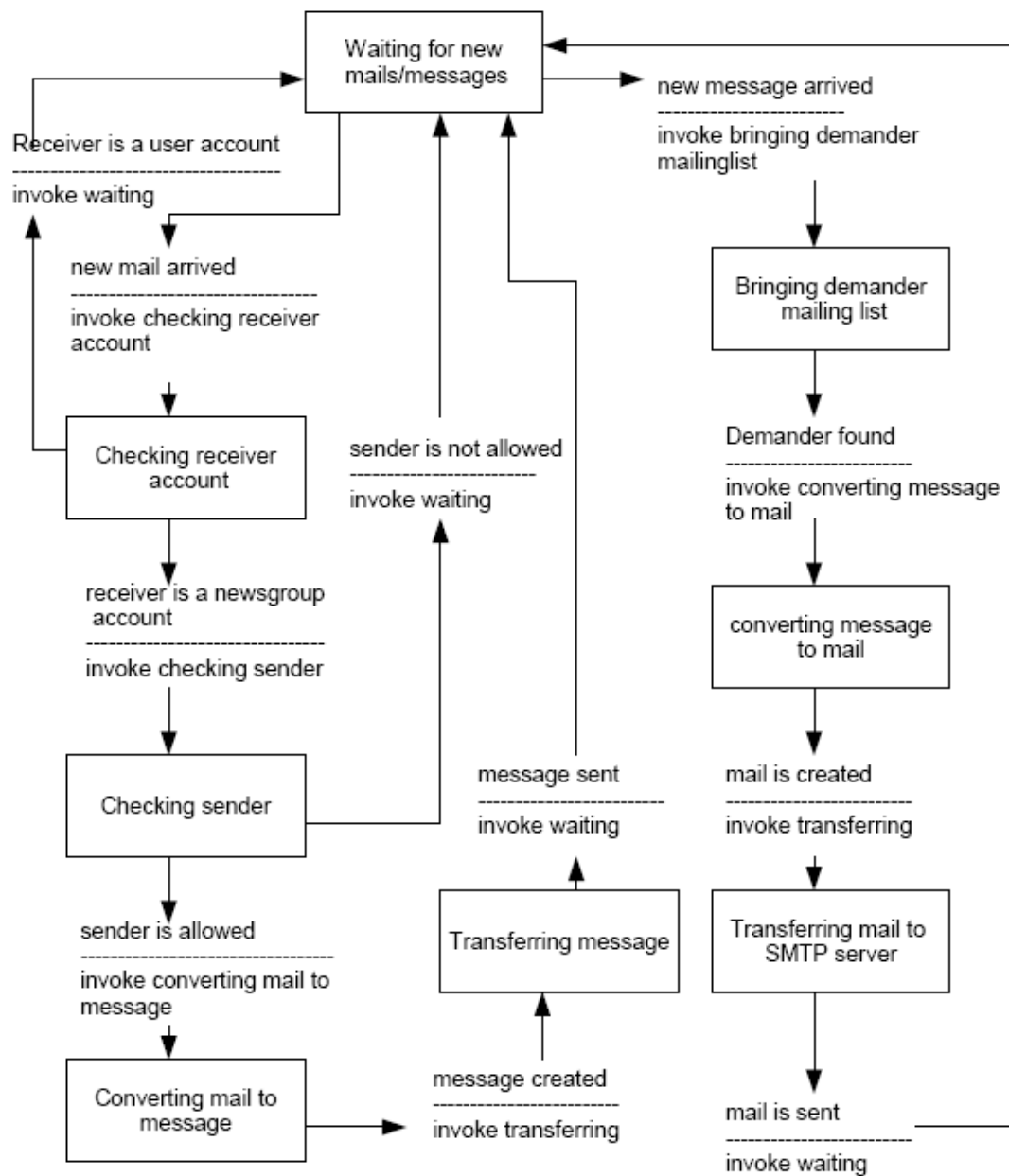
3.4 Mail Transfer Module

This module mainly provides access to the core using email addresses. It communicates with SMTP Server for receiving and sending mails. As mentioned in web module explanation, users can add themselves to the mailing lists of newsgroups. Web access module functions based on those mailing lists.

. Users can send e-mail to the newsgroups whose mailing lists they are registered in. Mail transfer module gets this request and converts sent mail to the form of an NNTP message. Then the message is saved in INN's file system using the script of the core. If the user is one that is not in the mailing list of the newsgroup, a filtering mechanism is actuated and the mail is rejected.

Mail transfer module is also responsible from sending articles to the users in the mailing lists of newsgroups as email. For this purpose, when an article comes to a newsgroup, core's MailAlerting layer alerts mail transfer module that there has come a message to some newsgroup. Then that message is immediately taken from the INN's file system using the script of the core and converted to email by mail transfer module. Then it is sent to the users' email addresses registered in the 'MailingList' table of the database.

➤ State Transition Diagram for Mail Transfer Module:



3.5 RSS Module

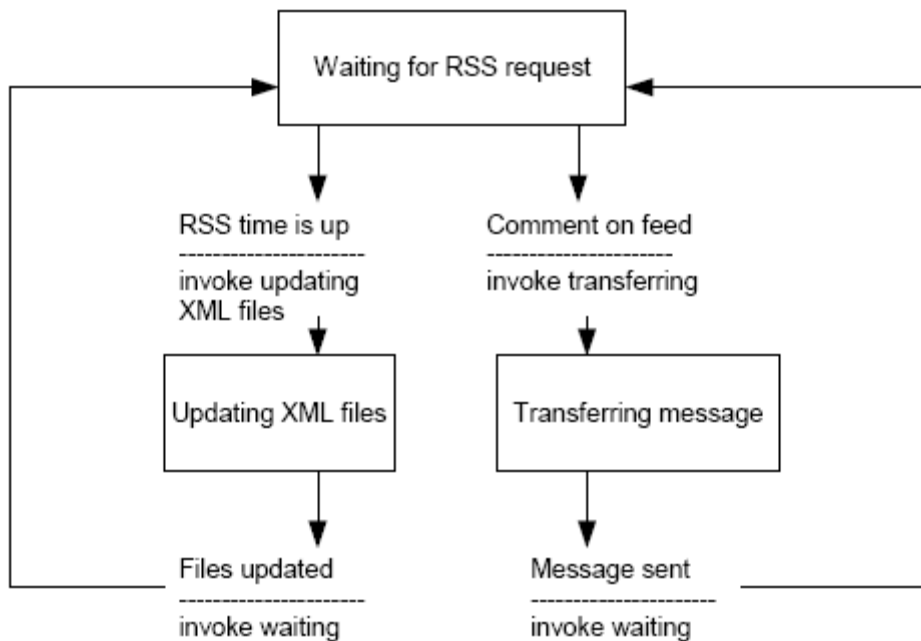
This module mainly provides access to the core using RSS Readers.

Some of the newsgroups are determined to provide RSS feeds while being created by the administrator. The time interval for generating XML files of newsgroups and under which directory those files would be kept is also determined by admin at that time. RSS module is responsible from generating those XML files regularly depending on this time interval and under the specified directory. The generated XML files can be seen by the user via the web.

Users can set their RSS Readers to get XML files of the newsgroups. It is ensured that only the newsgroups that the user has the permission to access are shown to the user via his/her RSS Reader.

Users can also make comments on RSS feeds. RSS Module is also responsible from taking those messages and saving them in INN's file system.

➤ State Transition Diagram for RSS Module:



3.6 System Management Module

System management module of our system is mainly responsible for administrative activities and it is directly related to system core. As mentioned before, there is an administrator of the system and he/she manages the system via the server machine. He/she uses a GUI, which is supplied by system management module, for the management of the entire system.

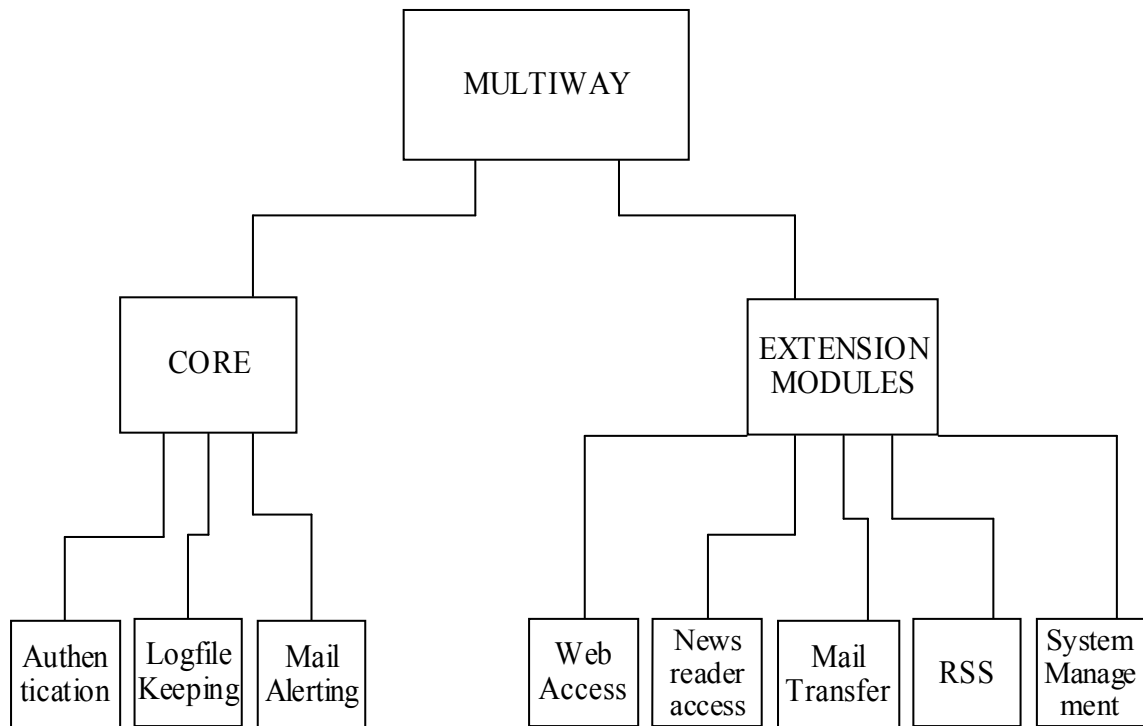
When admin creates a user account, the user information which is described in Data Design section is saved in 'User' table of the database. Username of the created user is generated by a script and sent to user's email address. When a user is created, he/she is assigned a limited quota to save some posted articles. Also, under which directory to keep the user's login history is specified by the administrator when a user is created.

When admin deletes a user account, all the data related to that user is deleted from the database. The articles posted by this user are also deleted from the file system of news server.

When admin creates a newsgroup, newsgroup information which is described in Data Design section will be saved in 'Newsgroup' table of database. Furthermore, NNTP Server INN in the core is informed of this new newsgroup. The newsgroup email account is also created. For file transferring and RSS feeding, new file directories for the newsgroup are also created automatically by the core.

When admin deletes a newsgroup, all the data related to that newsgroup is deleted. Deletion will be both in file storage and database.

- Below is the visual representation of the system structure using a Structure Chart:

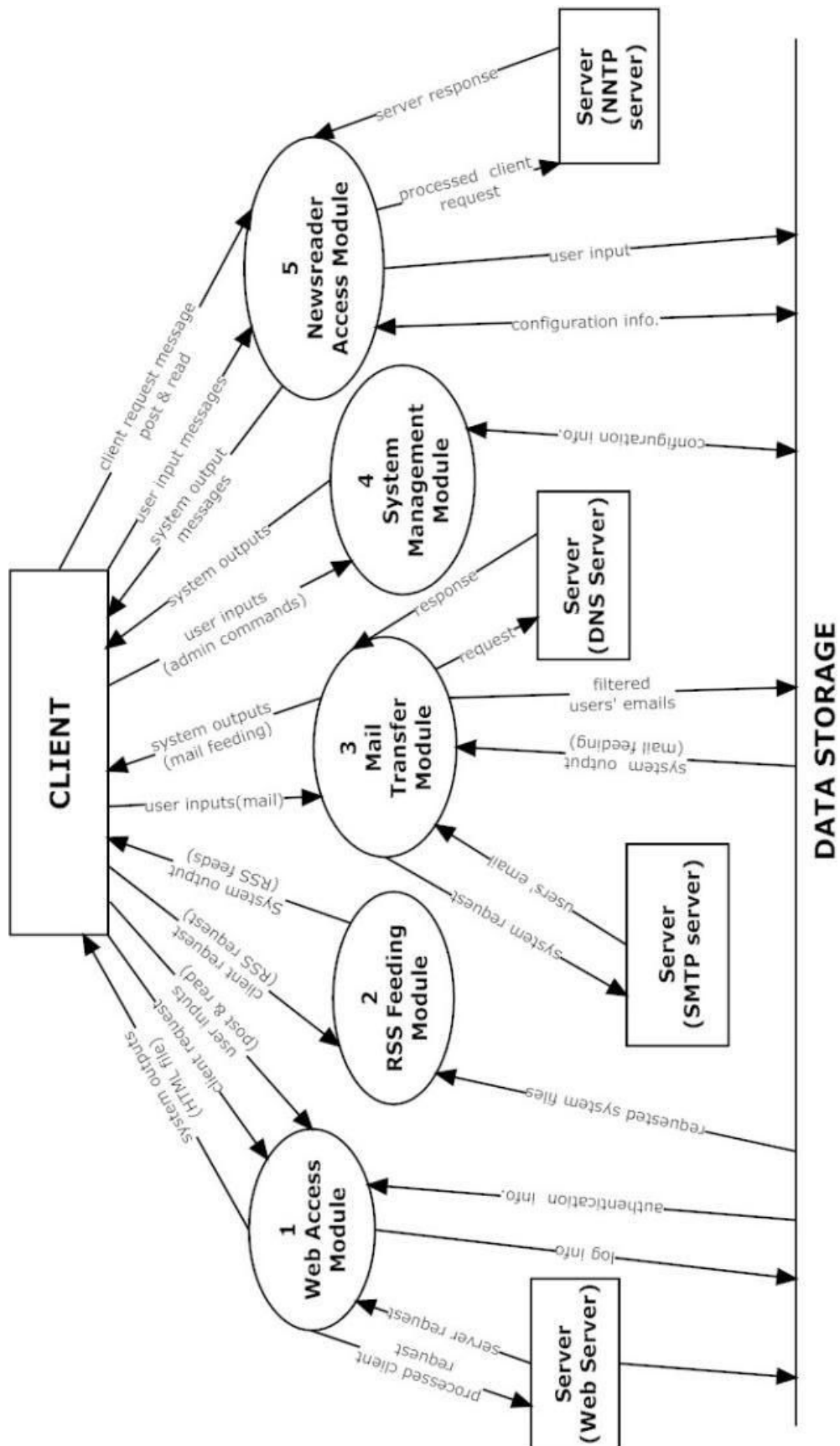


4. SYSTEM DATA FLOW

- Data Flow Level 0:

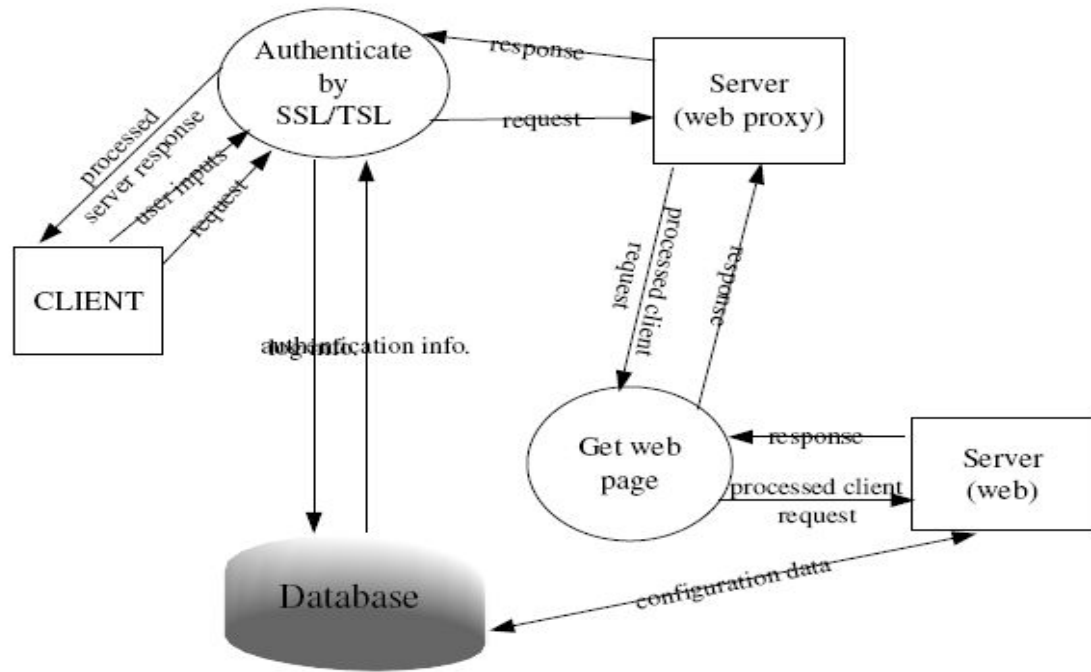


- Data Flow Level 1:

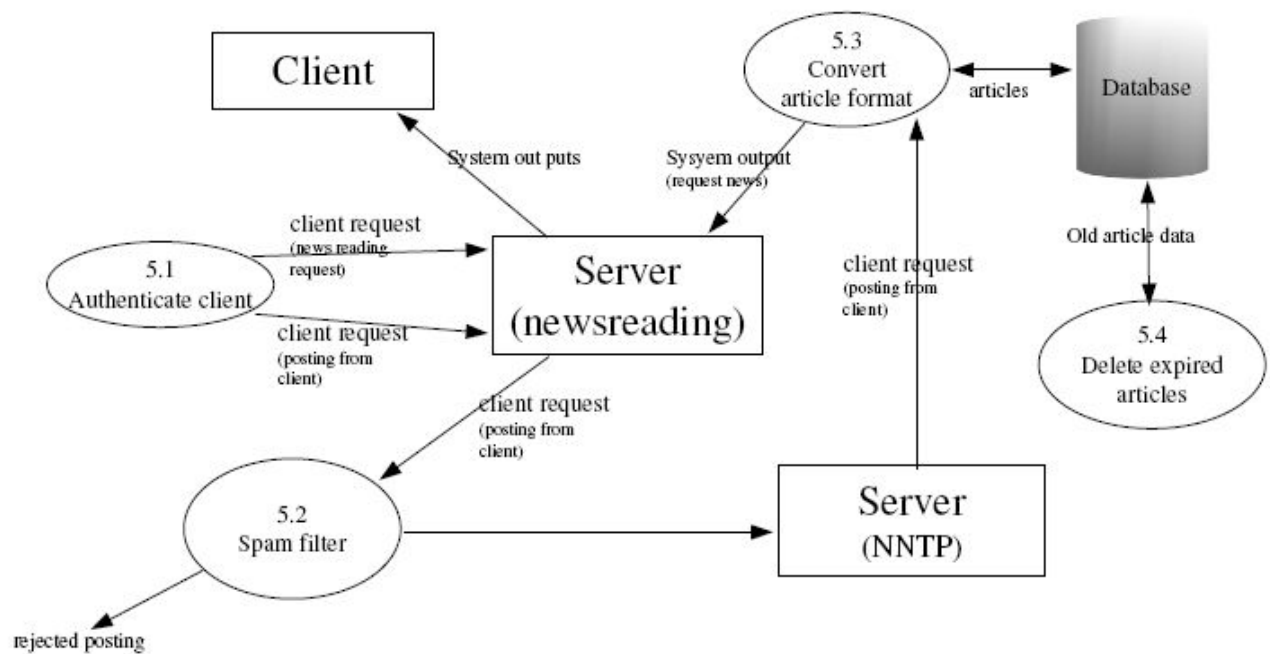


➤ Data Flow Level 2:

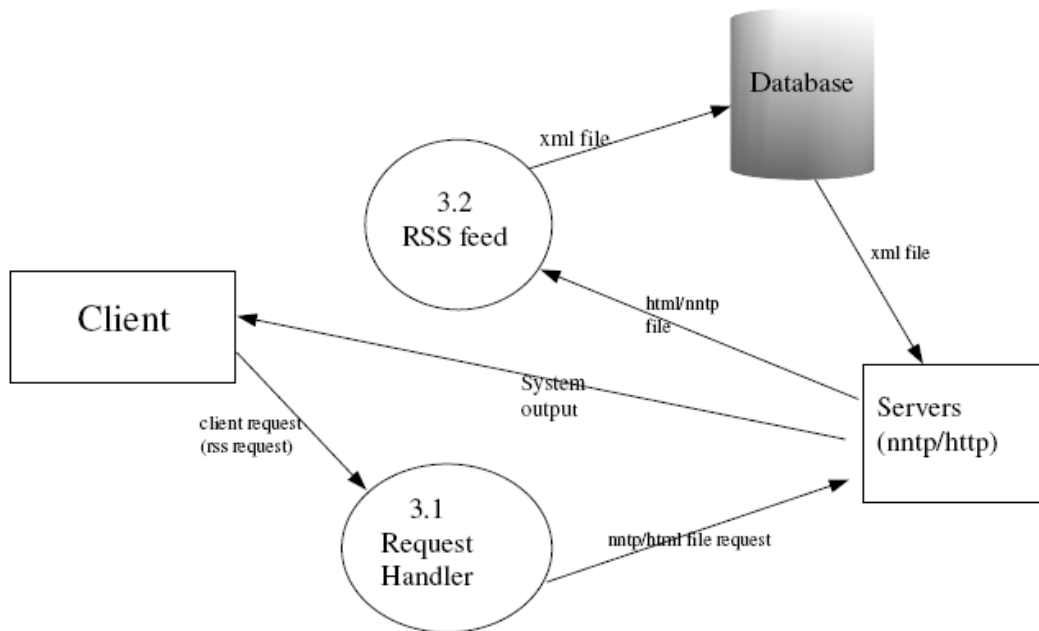
▪ **Web Access Module:**



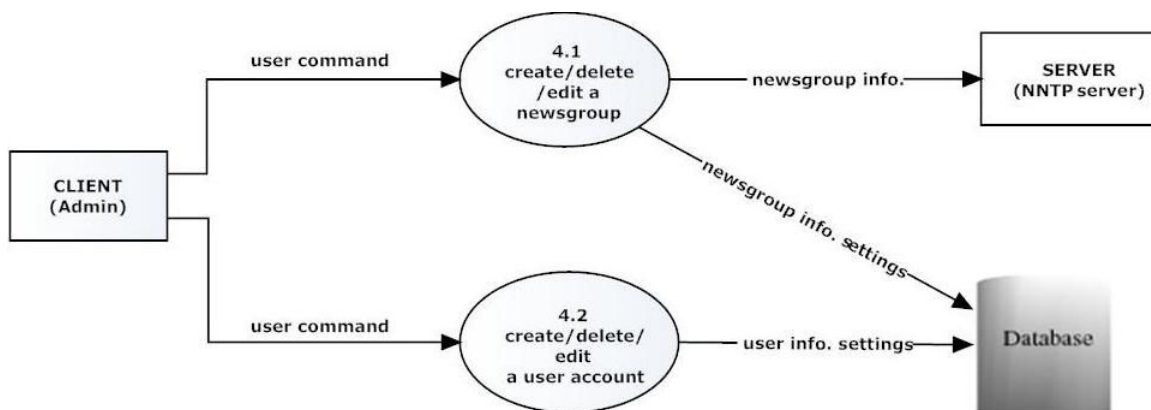
▪ **Newsreader Access Module:**



- ***RSS Module:***



- ***System Management Module:***



5. OBJECT-ORIENTED CONCEPTS

5.1 System Classes

There are seven classes in our system. Some of them are inherited from others. Every class has member fields and methods.

The 'User' class fields keep information about a user and methods of that class stand for the functionalities a user can perform. When a user logs in to the system after the authentication, an instance of 'User' class is created and its fields are set. Every time the user performs a task, the corresponding method of the created 'User' object is called.

The 'Admin' and 'Chief' classes are subclasses of 'User' class.

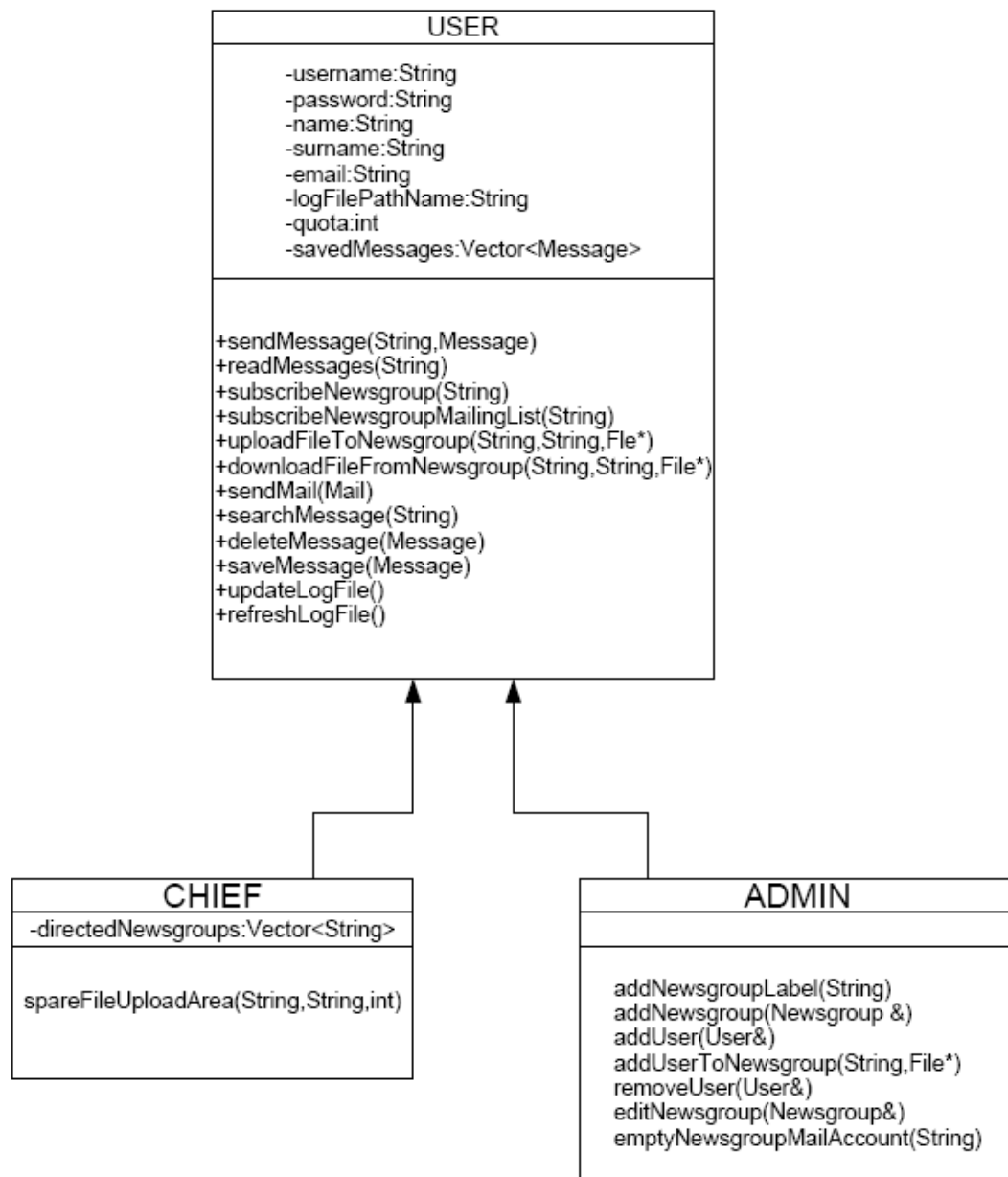
The 'Newsgroup' class fields keep information about a newsgroup and methods of that class stand for the system functionalities which affect newsgroups. An instance of 'Newsgroup' class is created whenever needed in the system and its fields are set. Every time some task affecting that newsgroup, the corresponding method of the created 'Newsgroup' object is called.

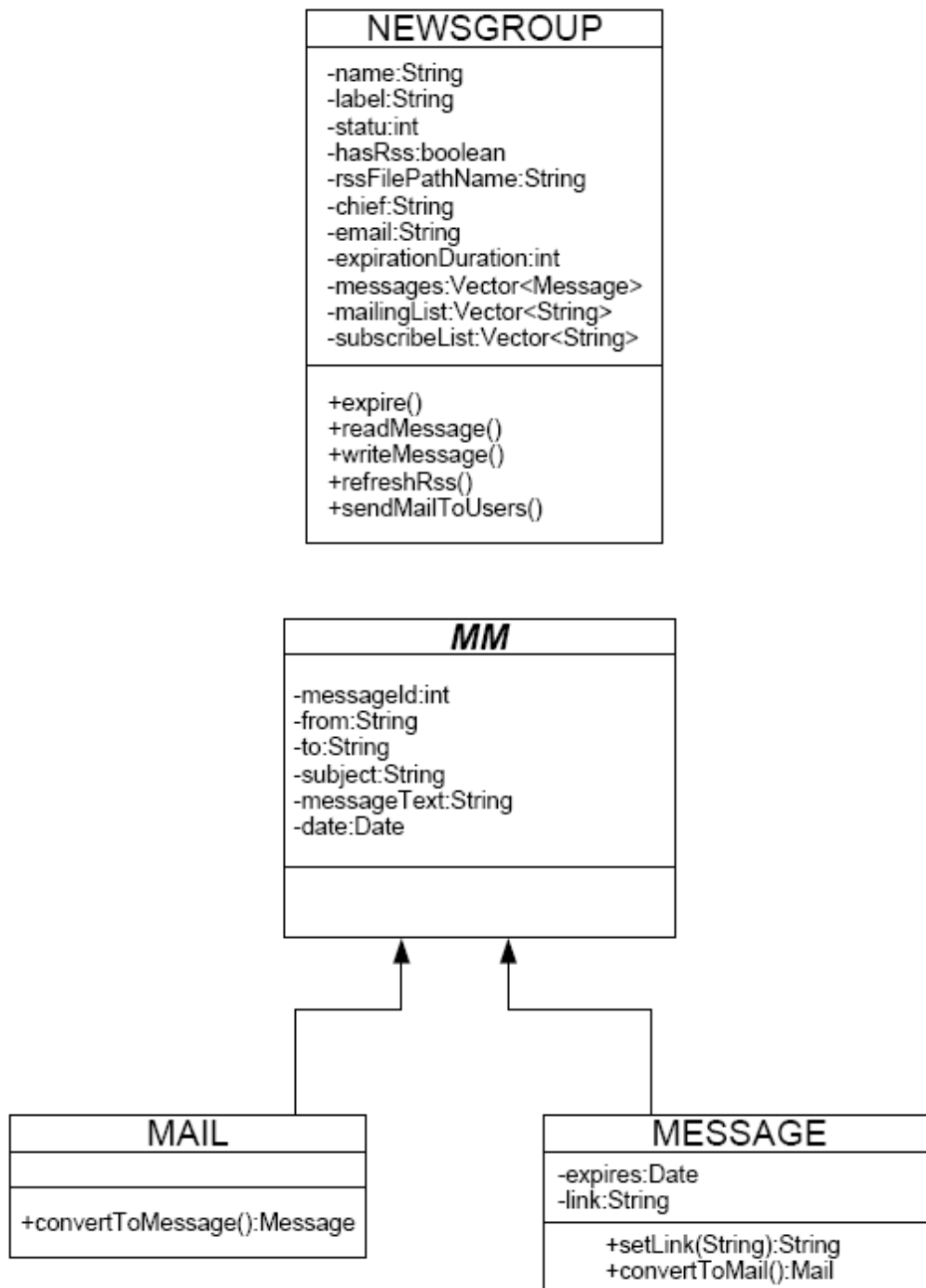
'MM' is an abstract class from which 'Message' and 'Mail' classes are inherited. A 'Mail' object refers to a mail article and a 'Message' object refers to an article which is in the format to be saved in INN's file system. Those objects are created whenever they are needed in the system and their methods are called for converting between message forms.

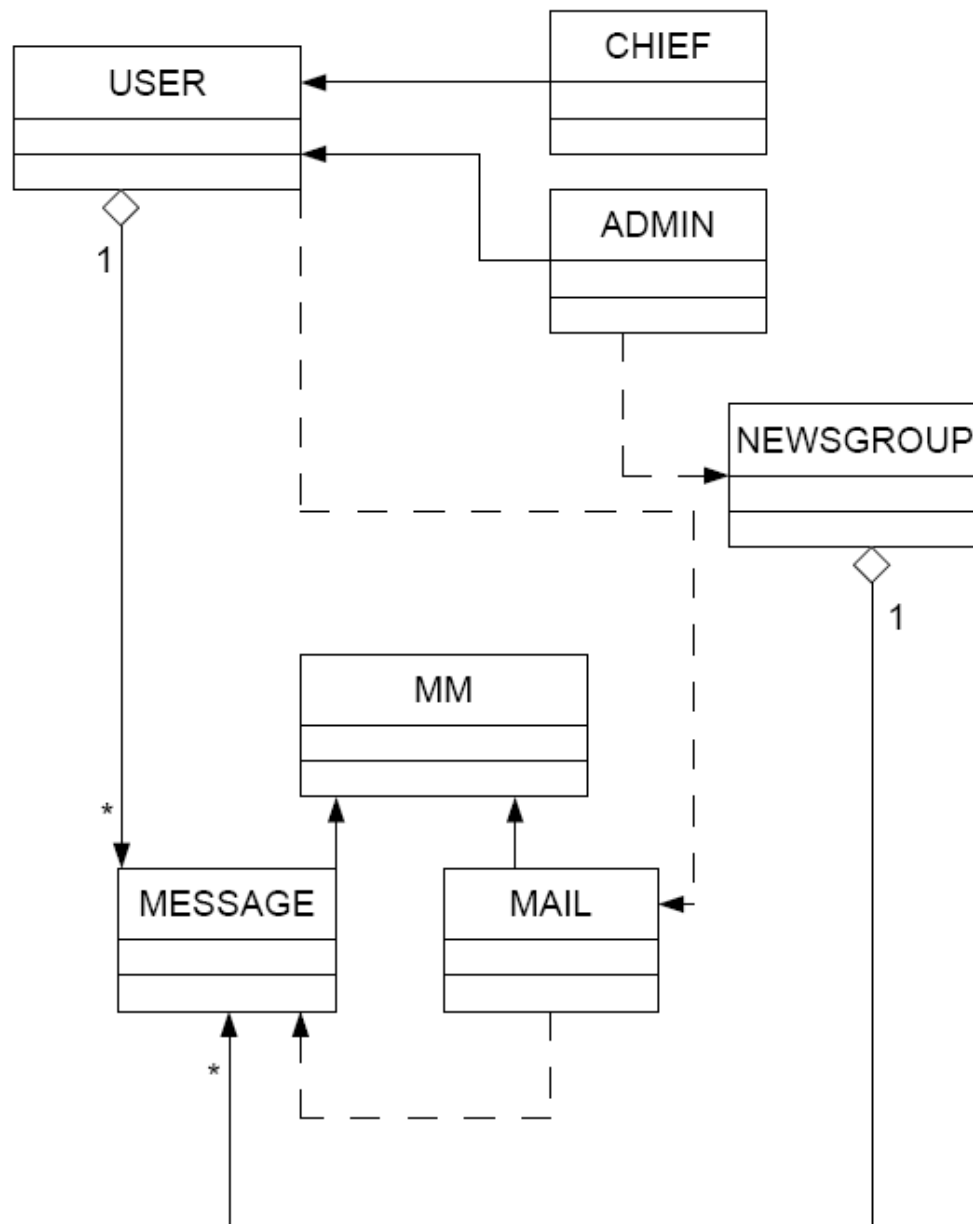
Because a 'Newsgroup' object has 'messages' field and so includes many messages sent, there is a 1 to many relation between classes 'Newsgroup' and 'Message'.

Similarly a user has saved messages in his/her account, so there is a 1 to many relation between classes 'User' and 'Message'.

Below are the class diagrams of the system in order to easily explain object-oriented design of the system by visual representations.



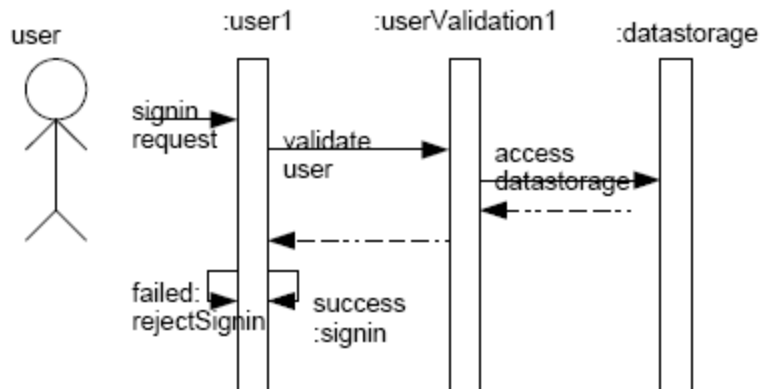




5.2 Interactions among Classes

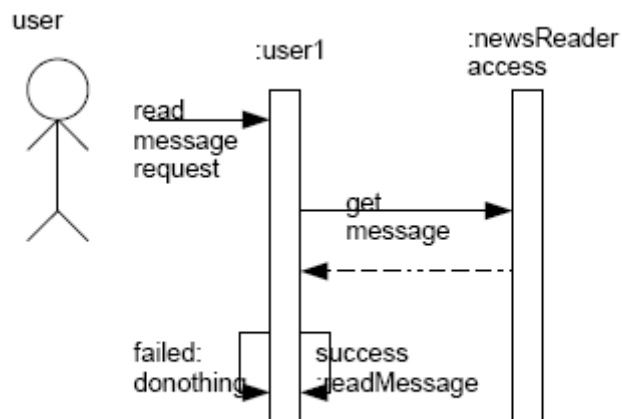
In this section, we represent interactions among system classes by considering the sequence of object method calls and message exchanges. We use sequence diagrams for visual representations.

➤ Sequence Diagram for User Login:



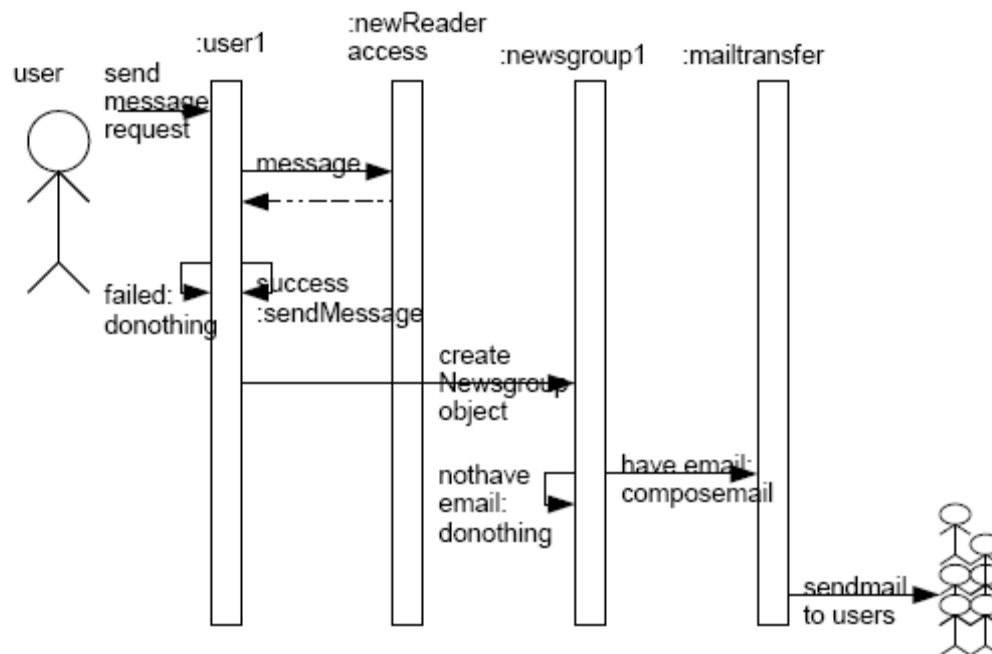
When user logs in to the system, he/she is checked for authentication using user information kept in the database. If user entered information matches with the database one, the user is logged into the system.

➤ Sequence Diagram for User Read Message:



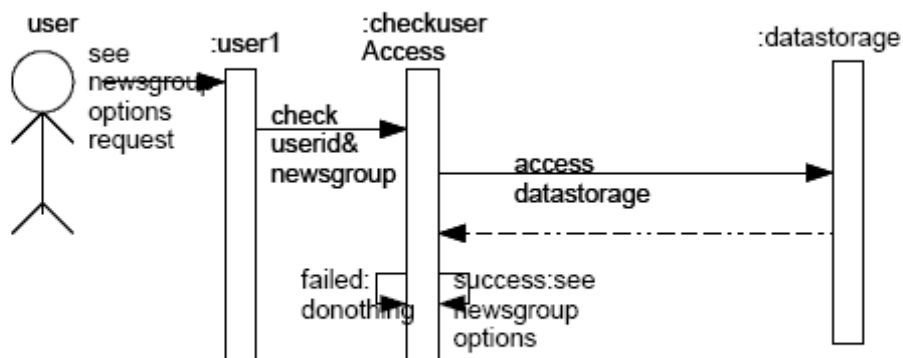
When user requests to read a message, article is brought from NNTP server to the user.

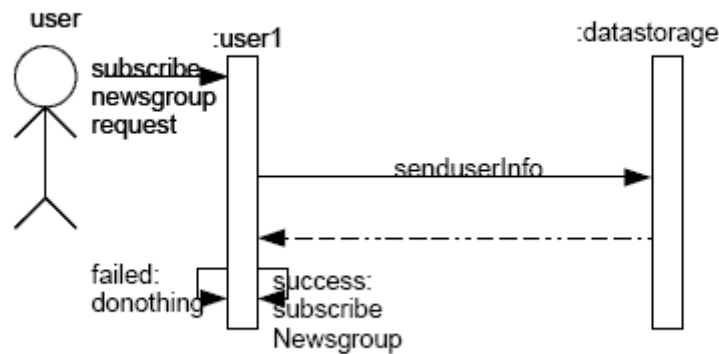
➤ Sequence Diagram for User Send Message:



When user requests to send a message to the system, article is put into NNTP Server and also sent to emails of users in the mailing list of the newsgroup.

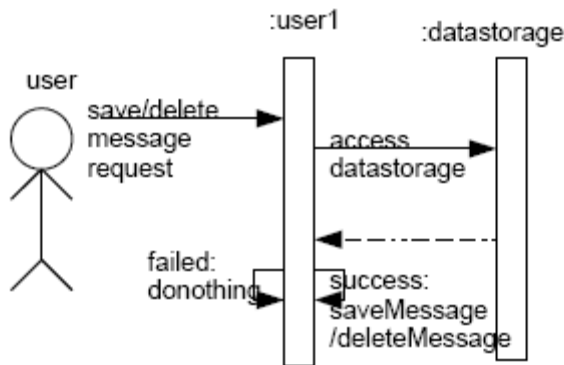
➤ Sequence Diagram for User Subscribe Newsgroup:





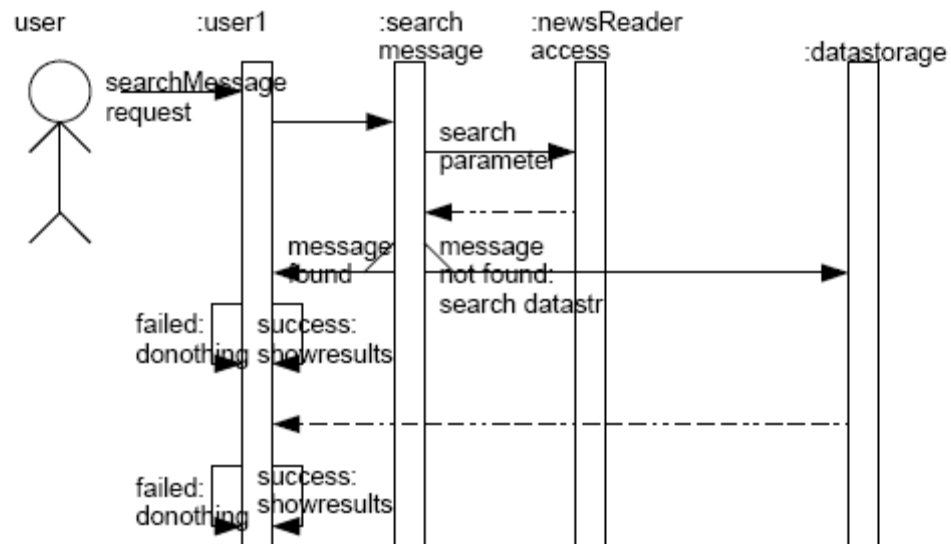
To subscribe to a newsgroup, that's, to see that newsgroup in his/her interface, the user must first see the allowed newsgroups he can subscribe to and then he selects one to subscribe. The subscription information is kept in the database.

➤ Sequence Diagram for User Save/Delete Message:



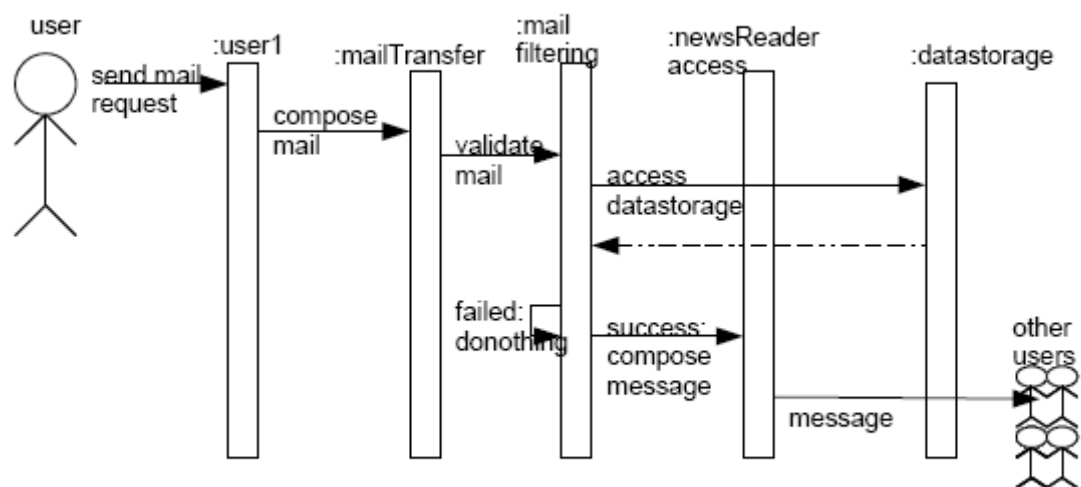
When the user requests to save a message in his/her account or remove a message from his/her interface, the system accesses to data storage and then the save or delete action is performed on the data.

➤ Sequence Diagram for User Search:



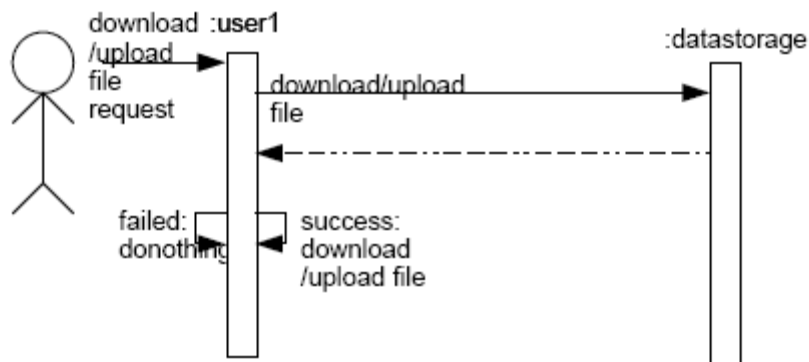
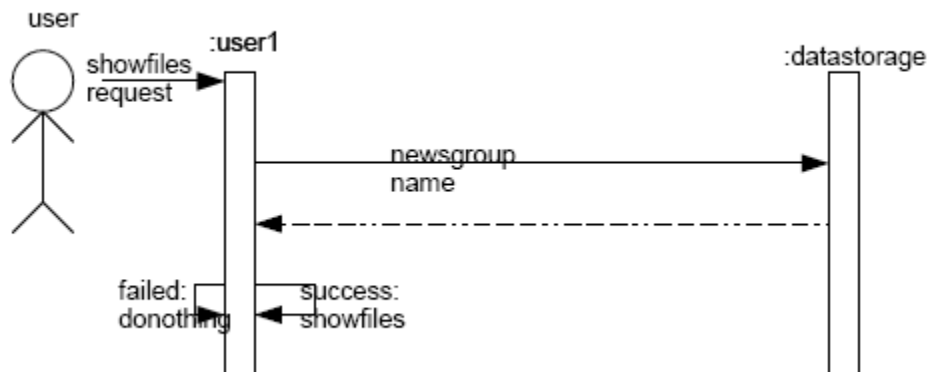
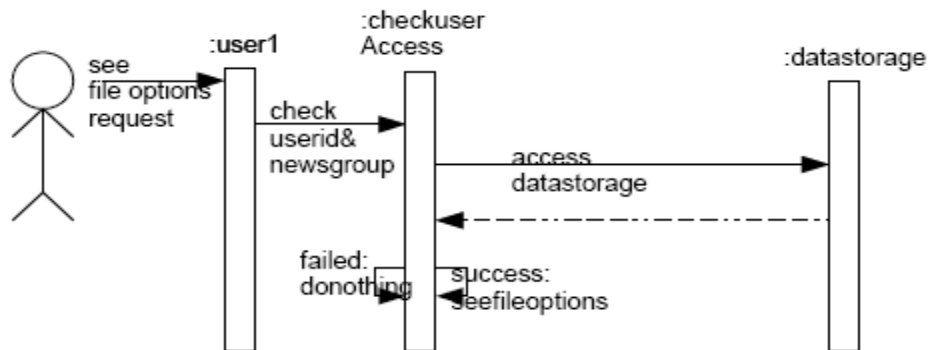
When the user searches for the articles for some key, the entire INN file system and archive file system are searched. Matching messages are shown to the user.

➤ Sequence Diagram for User Send Email:



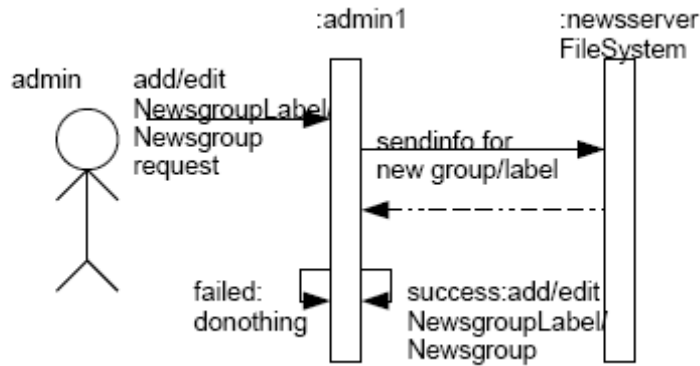
When the user wants to send a message to a newsgroup via email, the user is checked if he/she is in the mailing list of the newsgroup. If so, mail is converted to message and saved in the data storage.

➤ Sequence Diagram for User Upload/Download Files:

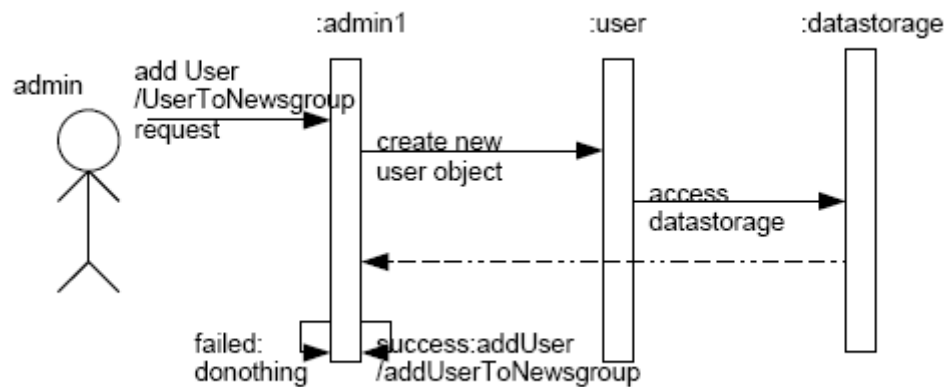


When a user wants to upload files to / download files from the system, he/she must first see the available files of the allowed newsgroups and then the desired operation is performed.

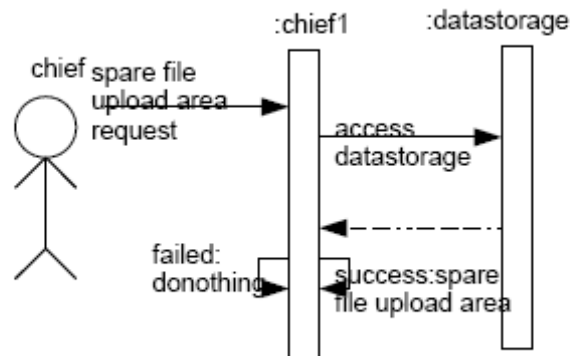
➤ Sequence Diagram for Admin Create Newsgroup:



➤ Sequence Diagram for Admin Create User:



➤ Sequence Diagram for Chief Spare File Upload Area:



6. SYNTAX SPECIFICATIONS

In our MULTIWAY project, we use more than one language. But as it is explained in the introduction part, we use mainly Java, which is an object-oriented language. And as a part of standardization process for rapid development of our project, we have decided to develop special naming conventions. In addition, we have decided the syntax of descriptive comments that will be used for understandability and maintainability of our product. The details of the specifications are described below.

We have decided that all names should be comprehensible. For names that are composed of more than one word, lower case/upper case characters will be used to distinguish between consecutive words

✓ ***Naming the Classes:***

All classes will have names beginning with a capital letter. The classes with more than one word will have a capital letter at the beginning of each word to make it easy to understand the functionality of the class.

✓ ***Naming the Class Attributes:***

Attributes will begin with a lower case letter. In case there are more words, they will be distinguished by underscores. “user_id” “subgroup_name” are valid class attribute examples.

✓ ***Naming the Class Methods:***

Methods will begin with a lower case letter. In case there are more words in the method name, these words will be distinguished by capital letters at the beginning of each word. “getUserId” and “addNewsgroup” are valid method names.

✓ ***Naming the Database Table:***

Names of the tables in the database will begin with capital letters and will continue with a capital letter for each consecutive word similar to the class conventions. Attributes of the tables will follow the naming convention for the class attributes; that is, will begin with lower case letters and continue with underscore before each new word.

✓ ***Naming the Files:***

Files that include the source code for a class will be named as the following respectively:

`<class_name><.java>`

✓ ***Naming the Global and Local Variables:***

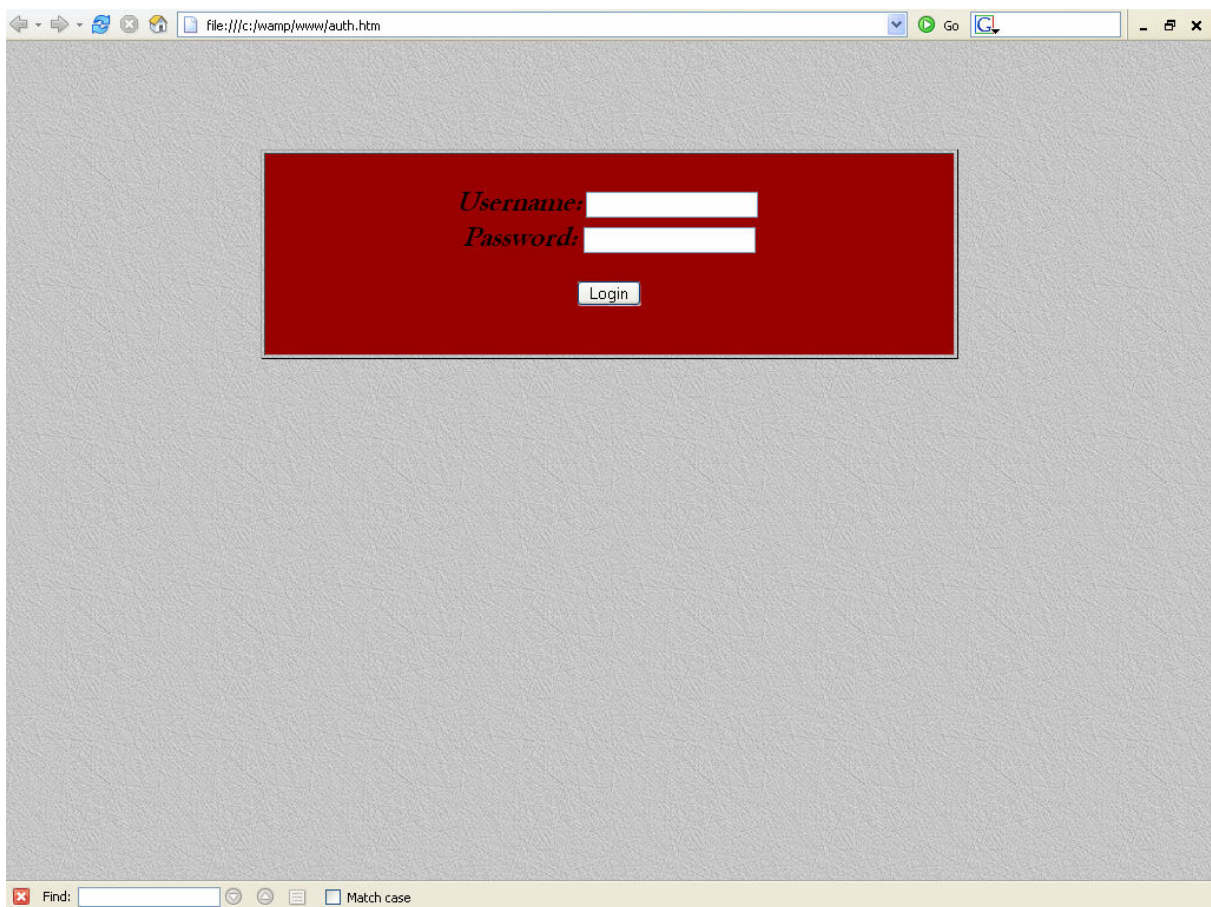
We will try to avoid using global variables as an appropriate software engineering principle. However, in case of any necessity, global variables will be prefixed with “g_”, since usage of global variables significantly decrease the understandability of the source code. For local variables, similar to global variables convention, the variables will be prefixed with “l_”.

7. USER INTERFACE DESIGN

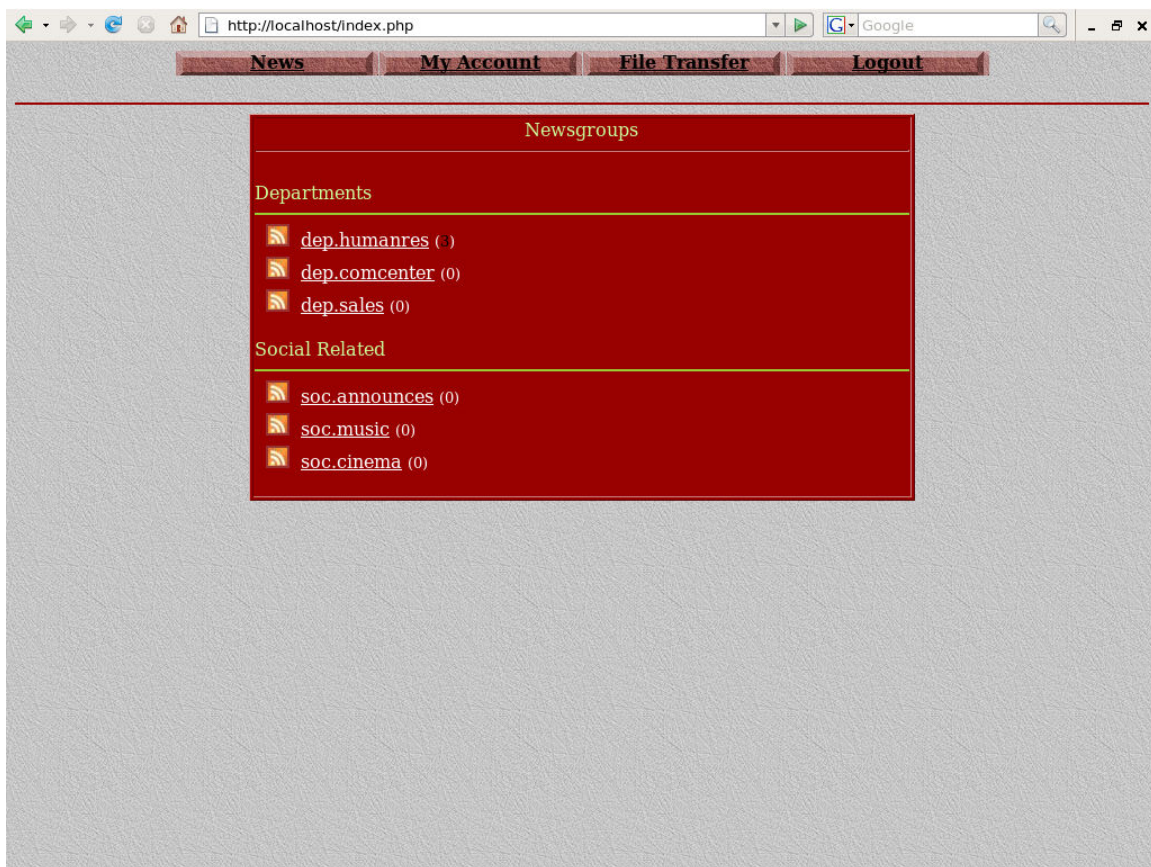
7.1 User Web Interface Design

In this section, user web interfaces are represented. Access methods other than web have interfaces themselves, namely newsreaders, RSS Readers and mail services. So, it is enough to create and present the web interfaces.

When the user enters the URL of the website of the system into his web browser, he faces with the login screen below. After entering his username and password, he presses login button. If his information is valid, he becomes an authenticated user.



After the user is authenticated, links of all the public newsgroups and the private newsgroups he/she is allowed to see are put on the interface as shown in the below screenshot. There are also RSS icons next to some newsgroup links. The user can see XML files of those newsgroups by clicking on the corresponding icon.



When the user clicks one of the newsgroups, he/she can see all the messages posted to that newsgroup as shown in the below screenshot. Date of sending, name of the sender and the subject are shown for a post and replies are represented in a tree structure.

Write a new message

DATE	SENDER	SUBJECT
16.01.	Alev Mutlu	Weekly meeting
16.01.	Aslı Kale	Re: Weekly meeting
16.01.	Safiye Çelik	Re: Weekly meeting

1

Newsgroups

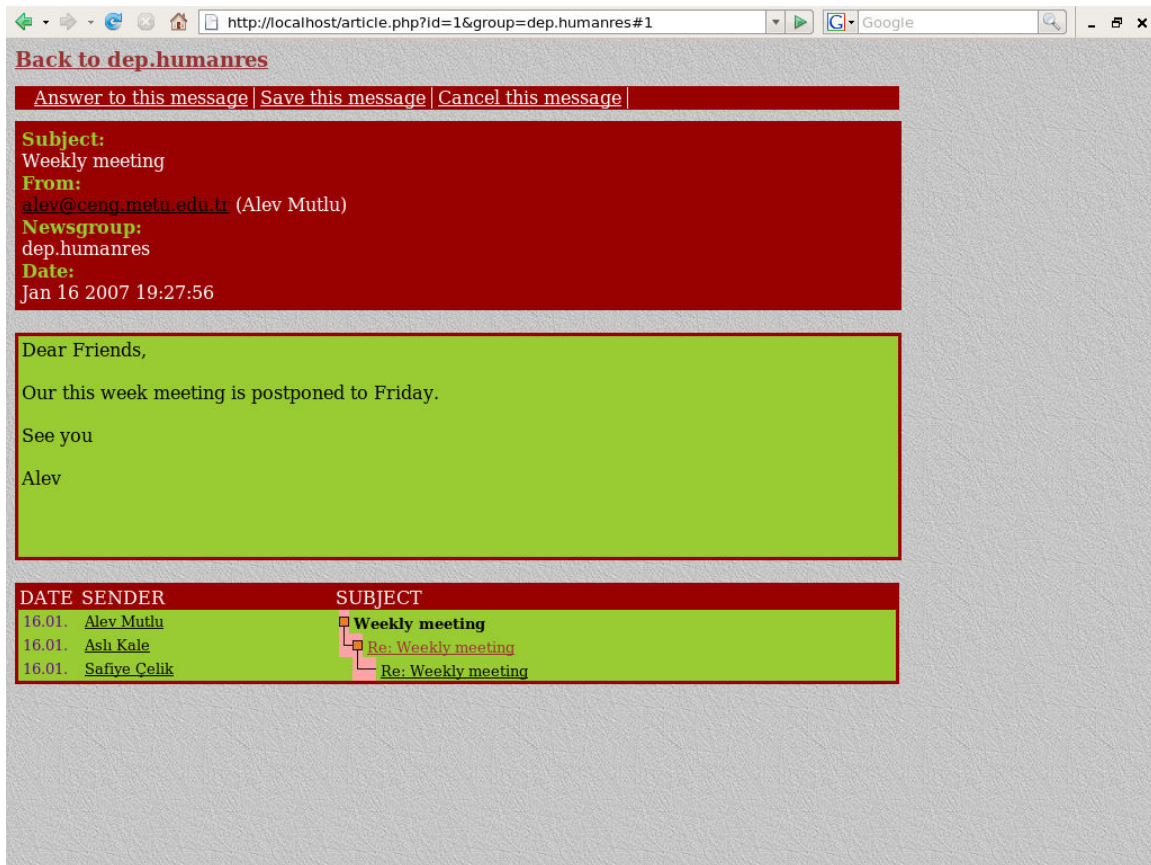
Departments

- [dep.humanres](#) (0)
- [dep.comcenter](#) (0)
- [dep.sales](#) (0)

Social Related

- [soc.announces](#) (0)
- [soc.music](#) (0)
- [soc.cinema](#) (0)

When the user clicks one of the posts, he/she faces with the below interface. In addition to the message text, subject of the message, sender of the message, sending date and the newsgroup the message is sent to are shown to the user. There is also the tree structure of that message and its replies in the interface. Options of answering the message, saving the message and canceling the message are also present.



[Back to dep.humanres](#)

[Answer to this message](#) | [Save this message](#) | [Cancel this message](#)

Subject:
Weekly meeting

From:
alev@ceng.metu.edu.tr (Alev Mutlu)

Newsgroup:
dep.humanres

Date:
Jan 16 2007 19:27:56

Dear Friends,

Our this week meeting is postponed to Friday.

See you

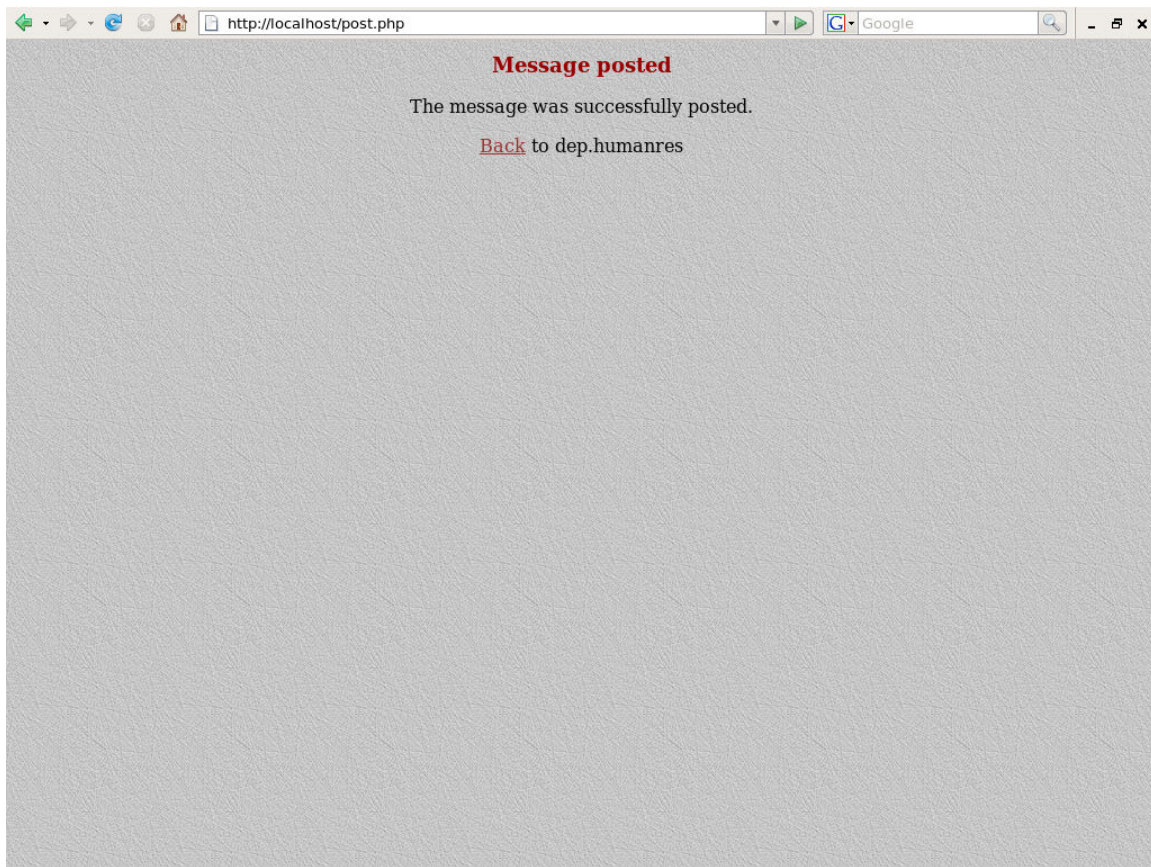
Alev

DATE	SENDER	SUBJECT
16.01.	Alev Mutlu	<input type="checkbox"/> Weekly meeting
16.01.	Ash Kale	<input type="checkbox"/> Re: Weekly meeting
16.01.	Safiye Celik	<input type="checkbox"/> Re: Weekly meeting

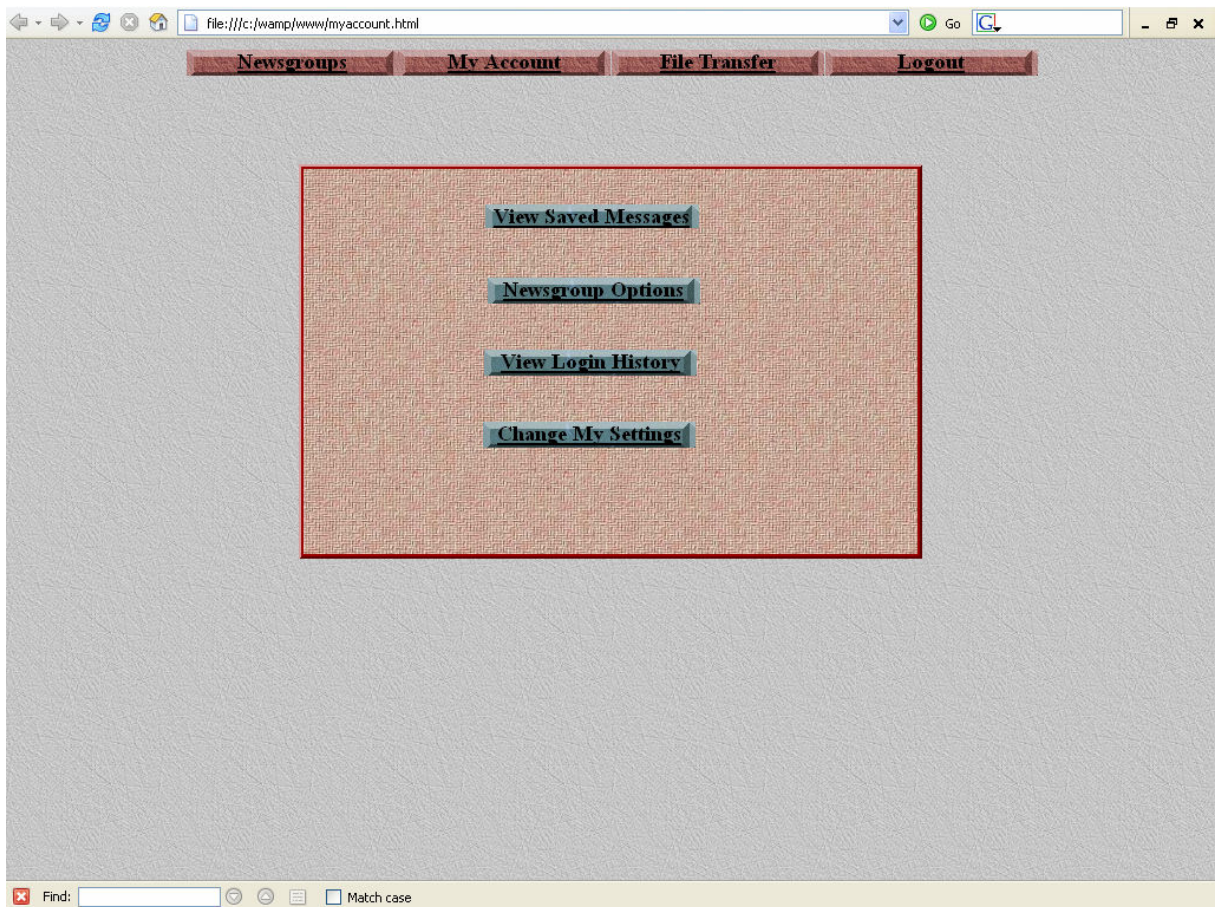
When the user presses 'Write a new message' or 'Reply to this message', he/she faces with the below interface. After writing the subject and the message text, he/she must click 'Send' button to send the message to the newsgroup.

The screenshot shows a web browser window with the address bar displaying `http://localhost/post.php?newsgroups=dep.humanres&type=new`. The browser's search bar contains the word "Google". The page has a grey background and a red header bar with the text "Back to dep.humanres" in red. Below the header, there is a red bar with the label "Subject:" in yellow, followed by a text input field containing "Weekly meeting". Below this, there is a red bar with the label "Message" in yellow. To the right of the "Message" label is a "Send" button. The message text area is a large white box containing the following text: "Dear Friends," "Our this week meeting is postponed to Friday." "See you" "Alev".

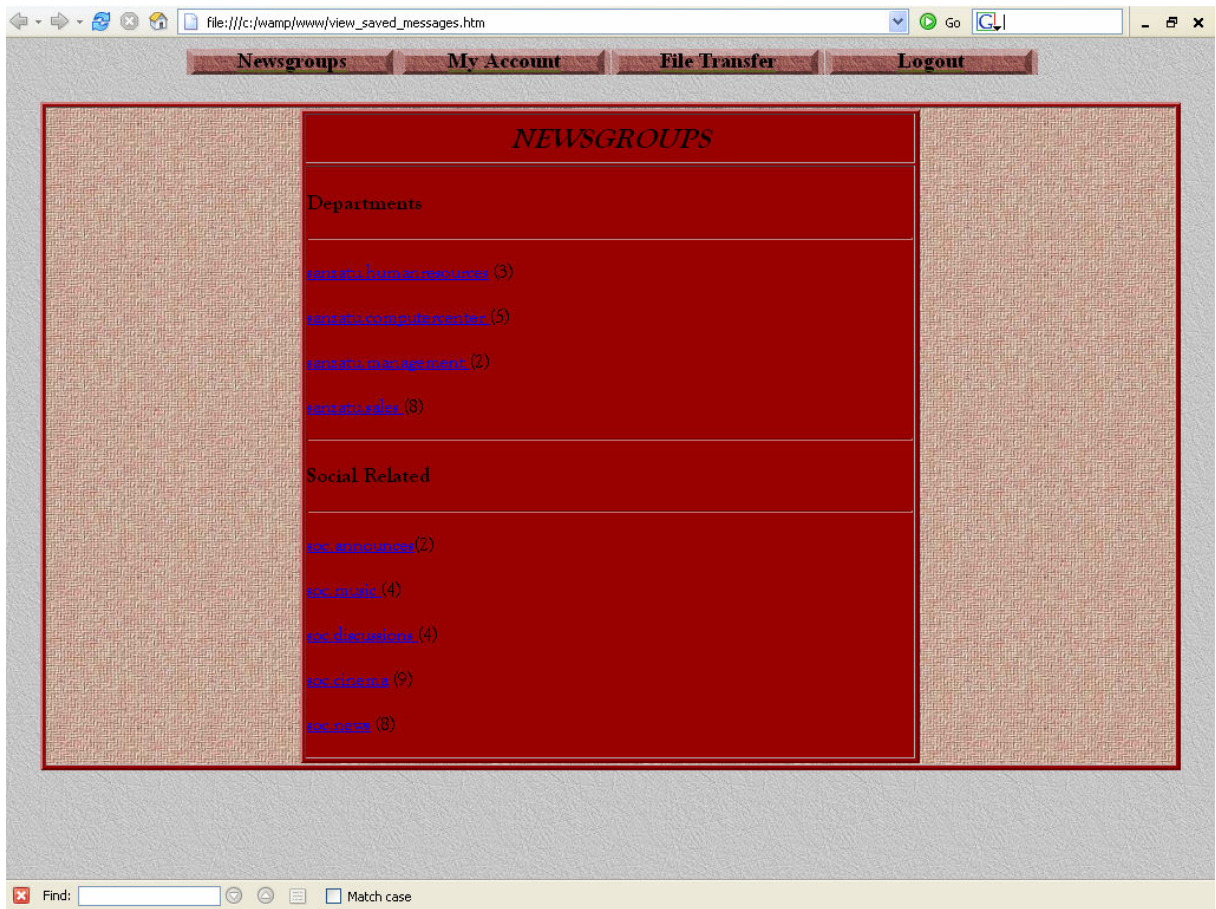
If the posted message is successfully kept in INN's file system, the below interface is presented to inform the user.



In addition to reading and writing messages, the user can manage his/her account on the system. When he/she presses 'My Account' button on the main menu, he faces with the below screen. Via his/her personal account, the user is able to view the messages he saved before, set his newsgroup options, view his login history and change his personal settings.



If the user presses 'View Saved Messages' button from his/her personal account interface, he/she faces with the below screen in which he sees the newsgroups whose messages he had saved before.



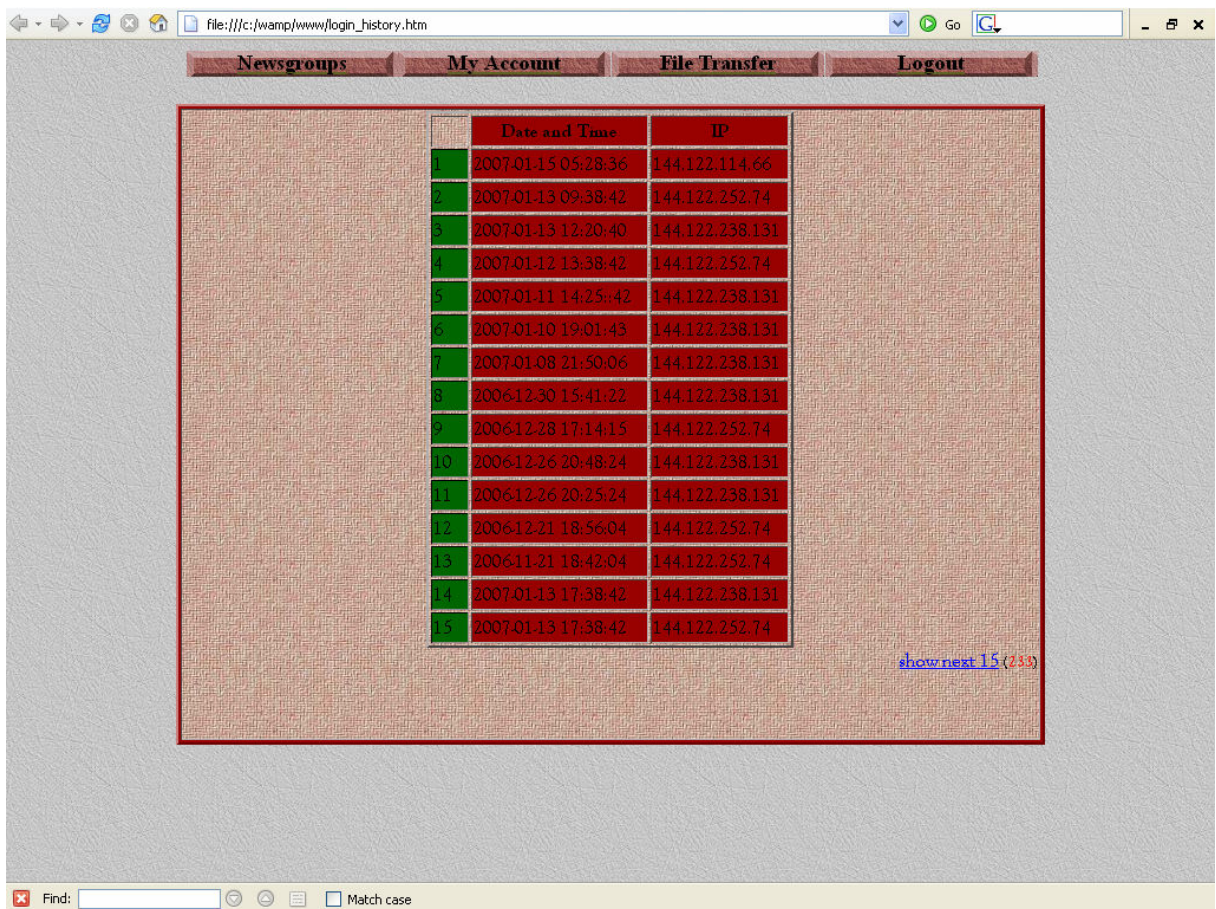
If the user presses on one of these newsgroups, he/she is forwarded to the below page in which he/she can see all of the messages he/she has saved for that particular newsgroup. Using that interface, he/she can also select a saved message and delete it.



If the user presses 'Newsgroup Options' button from his/her personal account interface, he/she is faced with the below screen. Using that interface, he/she can configure his/her settings about newsgroups he wants to be visible in his/her interface and about newsgroups whose mailing list he/she wants to be registered to. In that screen, the user can only see the newsgroups which he has access right (private groups which he is not a member of is hidden.). After signing the checkboxes, the user can save the settings by pushing 'Save Settings' button.

NEWSGROUPS	add to my interface	add to mailing list
Departments		
sanzatu.humanresources	<input type="checkbox"/>	<input type="checkbox"/>
sanzatu.computercenter	<input type="checkbox"/>	<input type="checkbox"/>
sanzatu.management	<input type="checkbox"/>	<input type="checkbox"/>
sanzatu.sales	<input type="checkbox"/>	<input type="checkbox"/>
Social Related		
soc.announces	<input type="checkbox"/>	<input type="checkbox"/>
soc.music	<input type="checkbox"/>	<input type="checkbox"/>
soc.discussions	<input type="checkbox"/>	<input type="checkbox"/>
soc.cinema	<input type="checkbox"/>	<input type="checkbox"/>

If user presses 'View Login History' button from his personal account interface, the page below showing the date and IP information about the user's last logins appears.



	Date and Time	IP
1	2007-01-15 05:28:36	144.122.114.66
2	2007-01-13 09:38:42	144.122.252.74
3	2007-01-13 12:20:40	144.122.238.131
4	2007-01-12 13:38:42	144.122.252.74
5	2007-01-11 14:25:42	144.122.238.131
6	2007-01-10 19:01:43	144.122.238.131
7	2007-01-08 21:50:06	144.122.238.131
8	2006-12-30 15:41:22	144.122.238.131
9	2006-12-28 17:14:15	144.122.252.74
10	2006-12-26 20:48:24	144.122.238.131
11	2006-12-26 20:25:24	144.122.238.131
12	2006-12-21 18:56:04	144.122.252.74
13	2006-11-21 18:42:04	144.122.252.74
14	2007-01-13 17:38:42	144.122.238.131
15	2007-01-13 17:38:42	144.122.252.74

[show next 15 \(233\)](#)

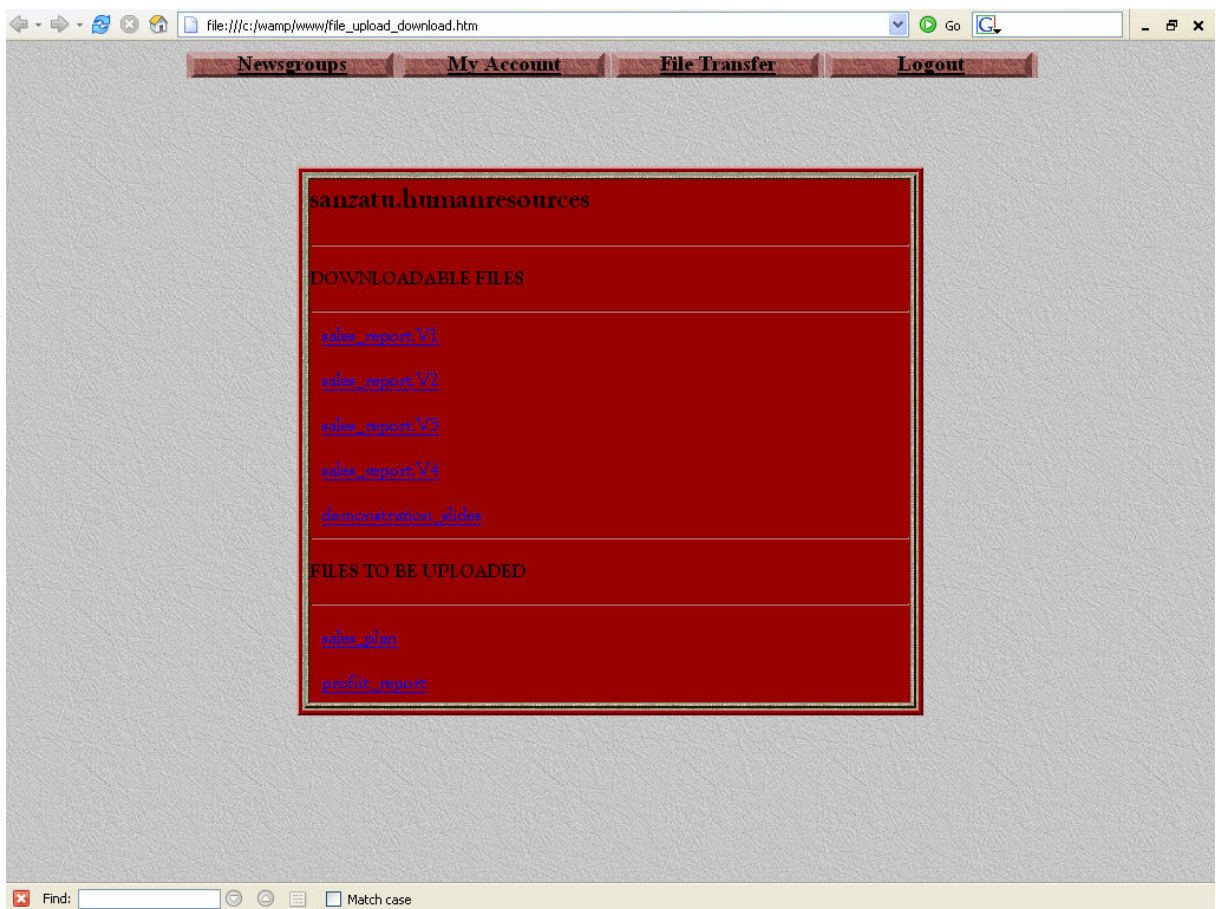
If user presses 'Change My Settings' button from his personal account interface, he faces with the below screen. From that window he can change his current username, password and e-mail address. System uses his username and password information for authentication purposes. The system uses the e-mail address which the user sets here when sending e-mails to users in the mailing lists of newsgroups.

The screenshot shows a web browser window with the address bar displaying 'file:///c:/wamp/www/change_my_settings.htm'. The browser has a menu bar with 'Newsgroups', 'My Account', 'File Transfer', and 'Logout'. The main content area is a red-bordered form with a light blue background. The form contains the following fields and buttons:

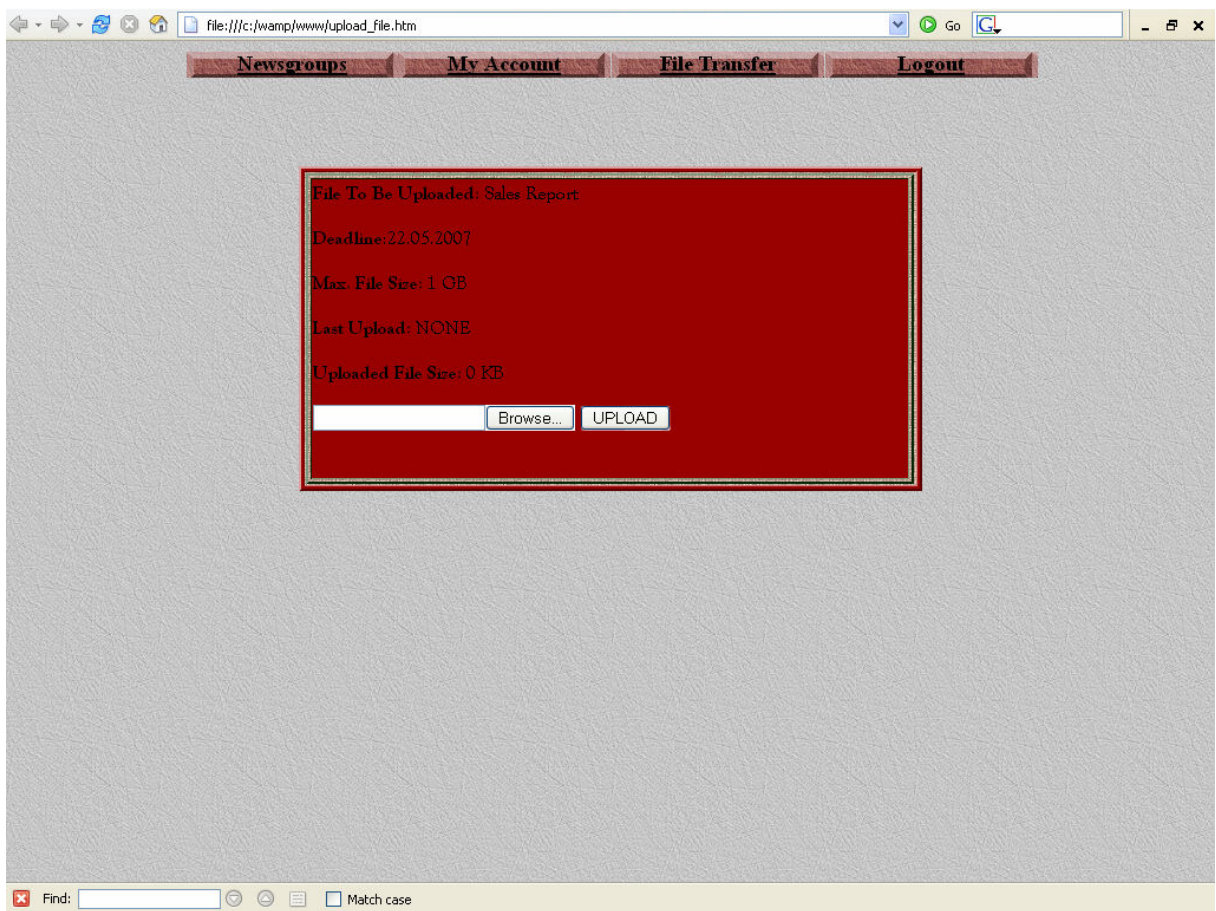
- Name Surname: Yılmaz Tokdemir
- Department: Human Resources
- Username: fgKul8M
- Current Password: [text input]
- New Password: [text input]
- Retype New Password: [text input]
- Change Password: [button]
- Email Address: yilmaz@gmail.com
- Save Settings: [button]

At the bottom of the browser window, there is a search bar with the text 'Find:' and a 'Match case' checkbox.

If user presses 'File Transfer' button, first he faces with the newsgroups which he is able to view. If he selects one of them, he sees an interface like the one below. Here he can view all the files he can download to his computer and the files which are required to be uploaded which are set by the newsgroup chief.



If he wants to upload a file, he presses its link and the below screen is seen from which he is able to upload his file. Here he can also see the due date and maximum file size information about the file to be uploaded.



7.2 Administrator Interface Design



A screenshot of a web application window titled "Administration". The window has a blue header bar with a small icon on the left and standard window controls (minimize, maximize, close) on the right. The main content area is light beige and contains a login form. The form consists of two labels, "Username:" and "Password:", each followed by a white text input field. Below the input fields is a grey button with the text "Login" in black.

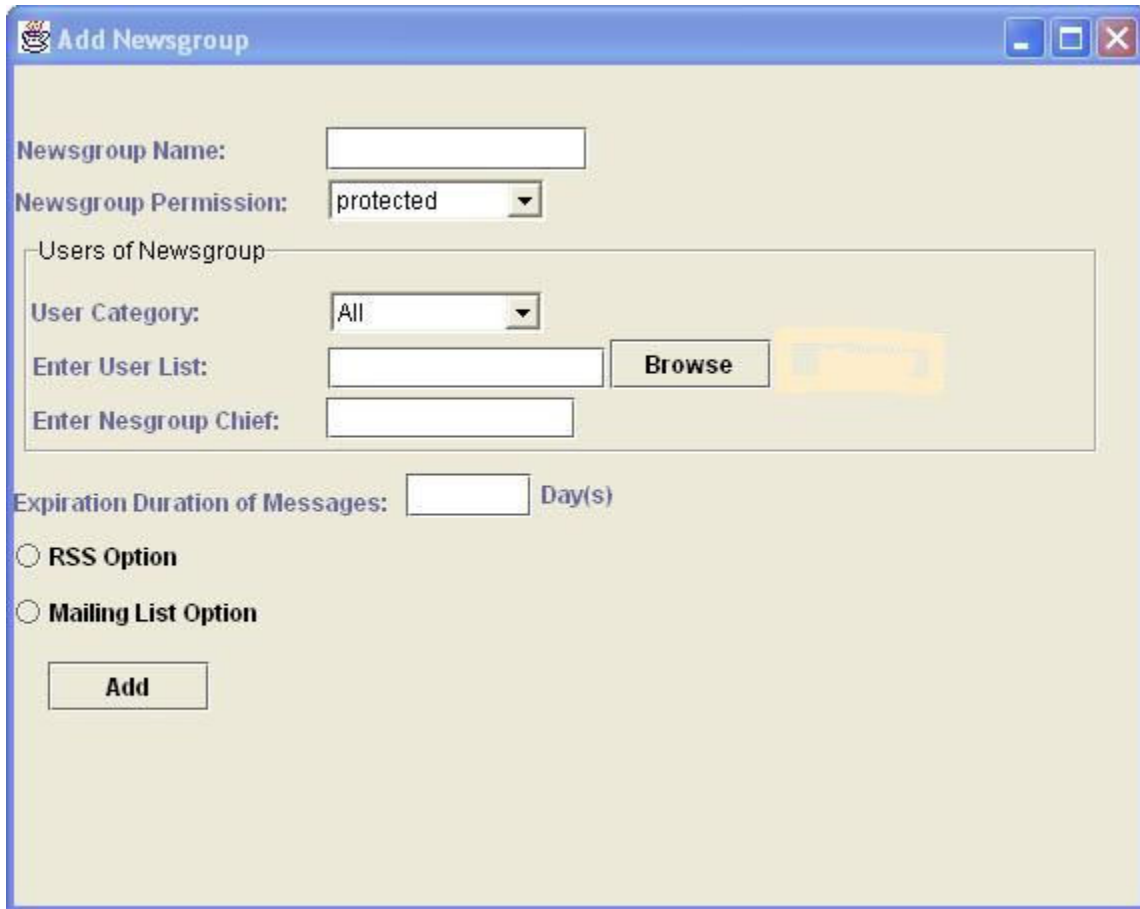
After logging in the system from the above login interface, the administrator encounters below 'Administration' interface.



A screenshot of the "Administration" interface after a successful login. The window has a blue header bar with the title "Administration" and window controls. The main content area has a light purple background. It features a vertical stack of six light beige rectangular buttons, each with black text. The buttons are labeled: "Add Newsgroup", "Edit Newsgroup", "Add User", "Delete User", "Change My Information", and "View System Log Files".

By using 'Administration' interface, admin can add a new newsgroup, edit an existing newsgroup, add a new user, delete an existing user, change his/her personal information and view system log files.

After pressing 'Add Newsgroup' button, s/he meets below 'Add Newsgroup' interface.

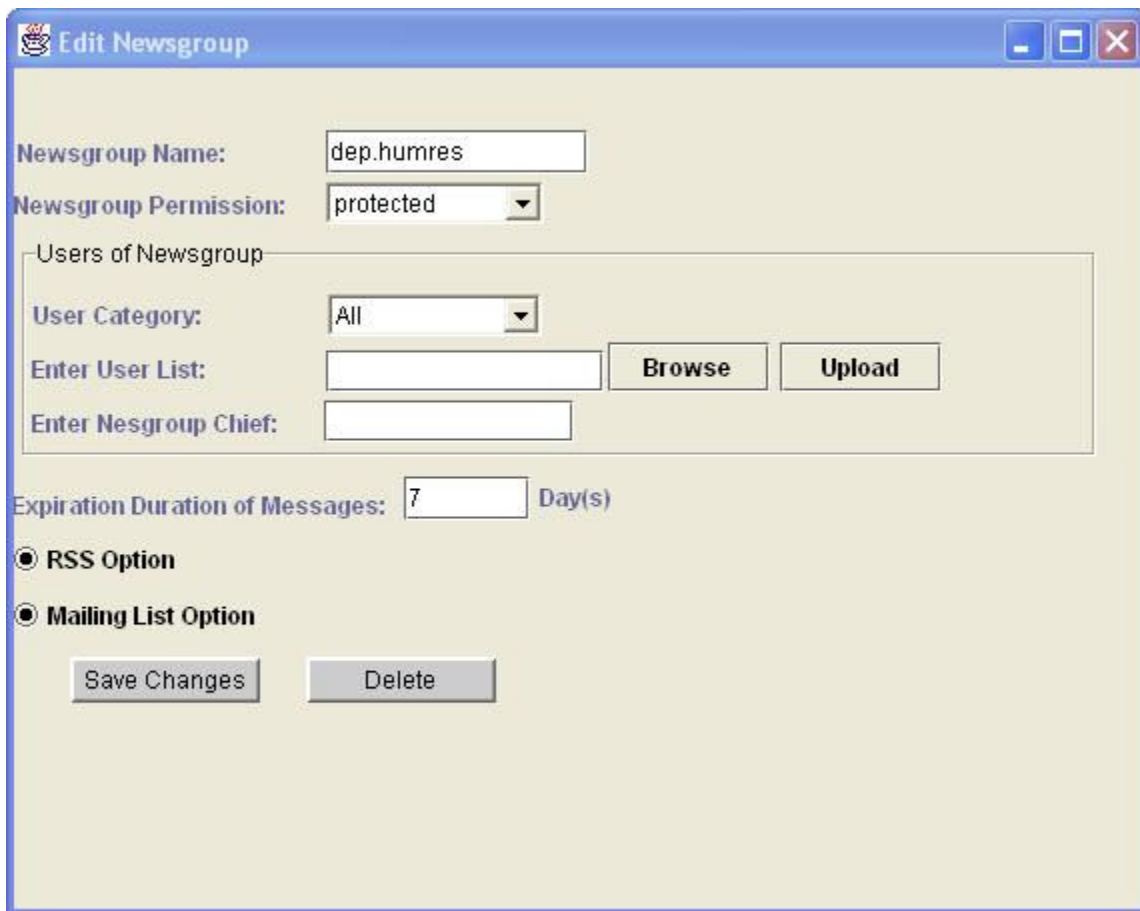


The screenshot shows a web-based interface for adding a newsgroup. The window is titled "Add Newsgroup". The form includes the following elements:

- Newsgroup Name:** A text input field.
- Newsgroup Permission:** A dropdown menu currently set to "protected".
- Users of Newsgroup:** A section containing:
 - User Category:** A dropdown menu currently set to "All".
 - Enter User List:** A text input field next to a "Browse" button.
 - Enter Nesgroup Chief:** A text input field.
- Expiration Duration of Messages:** A text input field followed by "Day(s)".
- Options:** Two radio buttons labeled "RSS Option" and "Mailing List Option".
- Add:** A button at the bottom left of the form.

For adding a new newsgroup to the system, s/he firstly enters newsgroup name (for example, dep.humres) and its permission. For newsgroups which can be seen by all subscribed users, s/he selects 'public' option whereas for newsgroups which can be seen by a restricted part of the subscribed users, s/he selects 'private' option from the 'Newsgroup Permission' field. For specifying users of the newsgroup, s/he firstly selects 'User Category'. If the 'User Category' is selected as 'Restricted' then s/he has to enter 'User List'. Usernames of the users are entered to system from a file which can be included by 'Browse' button. If there is a newsgroup chief of the private newsgroups, his/her username is entered. Expiration duration of the newsgroup messages should be entered. If this new newsgroup has an RSS or mailing list option, the administrator selects them from the interface. At the end, s/he can add newsgroup by pressing 'Add' button.

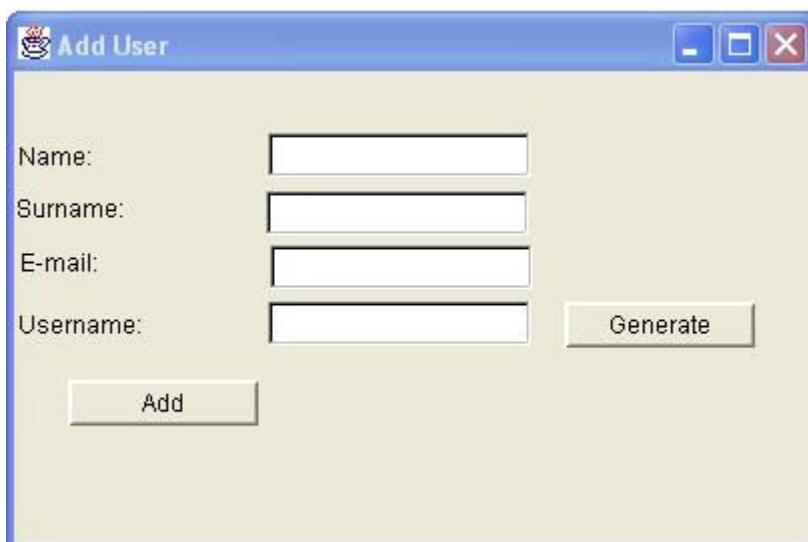
For editing an existing newsgroup, s/he updates its properties from below interface.



The 'Edit Newsgroup' window features a title bar with a standard icon and window controls. The main area contains the following fields and controls:

- Newsgroup Name:** A text input field containing 'dep.humres'.
- Newsgroup Permission:** A dropdown menu currently set to 'protected'.
- Users of Newsgroup:** A section containing:
 - User Category:** A dropdown menu set to 'All'.
 - Enter User List:** A text input field with a 'Browse' button to its right.
 - Enter Nesgroup Chief:** A text input field.
 - Upload:** A button located to the right of the 'Enter User List' field.
- Expiration Duration of Messages:** A text input field with '7' and the label 'Day(s)'.
- RSS Option:** A radio button.
- Mailing List Option:** A radio button.
- Save Changes:** A button.
- Delete:** A button.

For adding a new user to the system, the administrator uses below interface which can be reached by pressing 'Add User' button in 'Administration' interface.

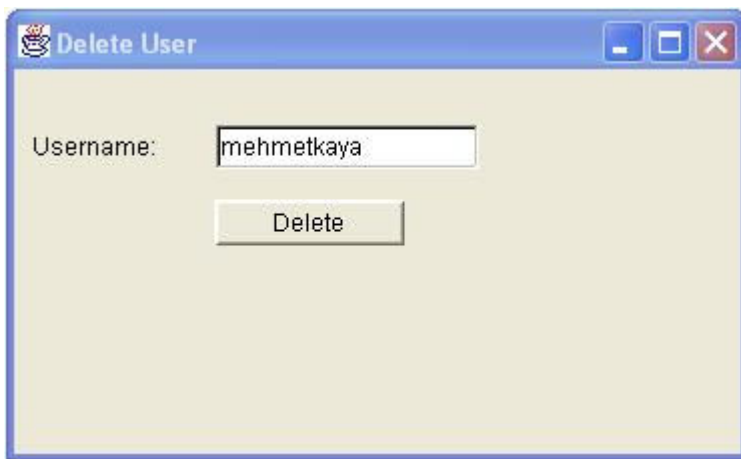


The 'Add User' window has a title bar with a standard icon and window controls. The main area contains the following fields and controls:

- Name:** A text input field.
- Surname:** A text input field.
- E-mail:** A text input field.
- Username:** A text input field.
- Generate:** A button located to the right of the 'Username' field.
- Add:** A button located at the bottom left.

The administrator enters name, surname and e-mail of the user. By pressing 'Generate' button, s/he can specify a unique username for the user. After these, s/he presses 'Add' button and our system will send username and password of the new user to the e-mail address of him/her.

The administrator can delete the user by entering username of him/her and pressing 'Delete' button from below interface.



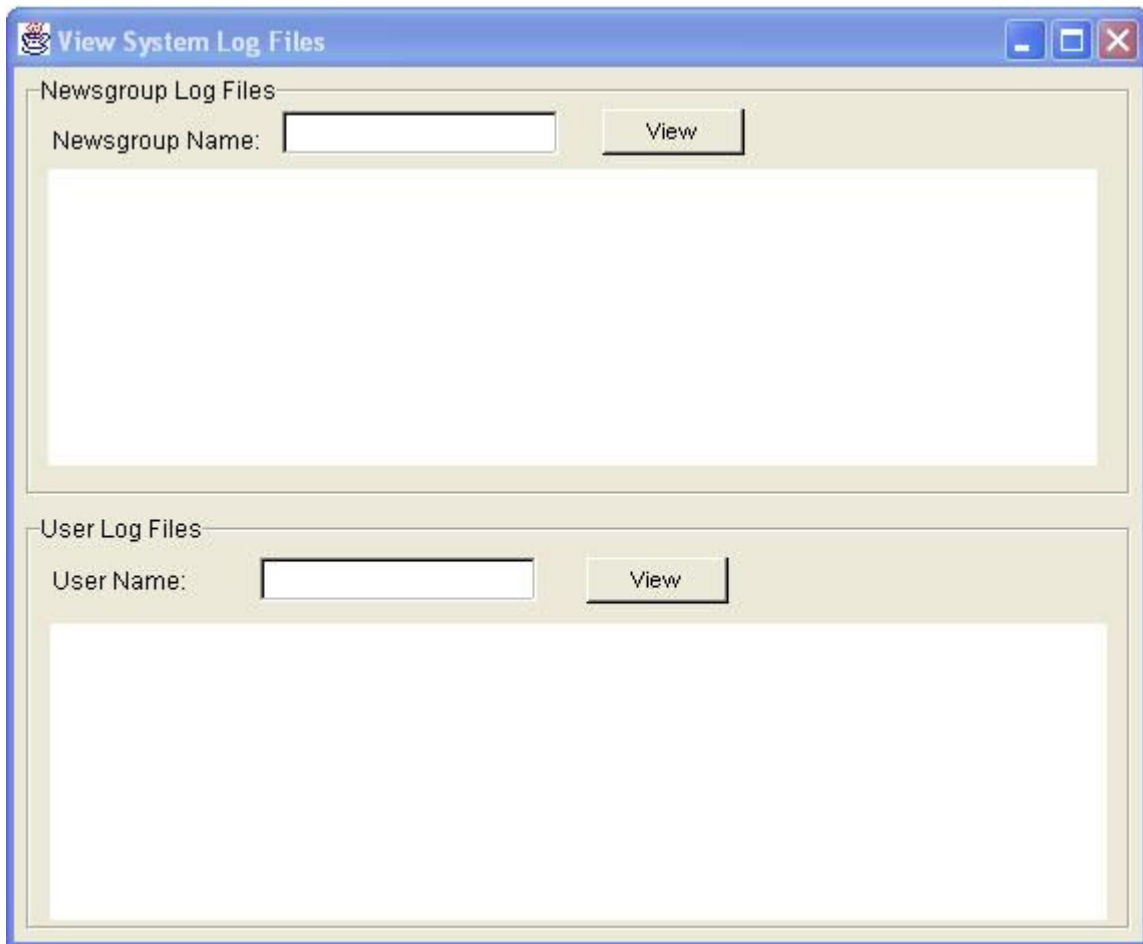
A screenshot of a web application window titled "Delete User". The window has a blue header bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main content area is light beige. It contains a label "Username:" followed by a text input field containing the text "mehmetkaya". Below the input field is a rectangular button labeled "Delete".

For changing his/her personal information, s/he can use below interface after pressing 'Change My Information' button in 'Administration' interface.



A screenshot of a web application window titled "Change My Information". The window has a blue header bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main content area is light beige. It contains five labels with corresponding text input fields: "Username:" with "emre", "Password:" with "*****", "Name:" with "Emre", "Surname:" with "Sogut", and "E-mail:" with "esogut@gmail.com". Below these fields is a rectangular button labeled "Save".

The administrator can view log file of the system by pressing ‘View System Log Files’ button in ‘Administration’ interface. Its interface is as follows:



The screenshot shows a window titled "View System Log Files" with a blue title bar and standard Windows window controls. The window is divided into two main sections. The top section is titled "Newsgroup Log Files" and contains a text input field labeled "Newsgroup Name:" followed by a "View" button. Below this is a large, empty rectangular area for displaying log data. The bottom section is titled "User Log Files" and contains a text input field labeled "User Name:" followed by a "View" button. Below this is another large, empty rectangular area for displaying log data.

Newsgroup log files and user log files can be seen from the text field by specifying their names and pressing ‘View’ button.

8. SYSTEM REQUIREMENTS

8.1 Minimal Hardware Requirements

Minimal hardware requirements for our project are: A PC with the following configuration will be needed:

>>Development

- _ Intel Pentium IV 2 GHz
- _ 1GB DDR RAM
- _ 100GB Hard Disk Space
- _ Local or Wide Area Network

>>End User

- _ Intel Pentium IV 1 GHz
- _ 512 MB DDR RAM
- _ 40GB Hard Disk Space
- _ Internet Connection

8.2 Minimal Software Requirements

Software requirements for the project are divided into categories: Development and End-user. For the Development part, whole the requirements of the open source servers that could be used in our project core are also considered.

• *Development*

- _ Linux Distribution such as Ubuntu 6.0 or OpenBSD 2.8 and up or FreeBSD 2.2.x and up etc
- _ C++ compiler (such as egcs, g++)
- _ GNU make
- _ Either the GDBM or Berkeley DB library
- _ Perl 5
- _ Python 1.5.2 or higher
- _ Postfix 2.0.16 or higher
- _ Mail-SpamAssassin 2.63 or higher

- _ Maildrop 1.7.0 or higher
- _ Cyrus - SASL 2.1.15 or higher
- _ Courier - authlib 0.58 or higher
- _ OpenSSL 0.9.7 or higher
- _ INN2
- _ Apache web server
- _ An implementation of yacc
- _ File system supporting FIFOs
- _ File system domain sockets as a must.

- ***End user***

- _ Recent Linux Distribution such as Ubuntu 6.0, or Windows XP (or newer) or Mac OS
- _ Web Browser (Mozilla Firefox, Microsoft Internet Explorer etc) or News Reader (Mozilla Thunderbird, OutlookExpress etc)
- _ Any RSS reader

9. PROJECT SCHEDULE

9.1 Project Task Set

For our project Multiway, scheduling activities and tasks are given below. Although we have not started the implementation phases according to our Gannt Chart yet, due to the prototype demo at the end of this semester, we have considered basic implementation issues.

- **Framework Activities**
 - Customer Communication ✓
 - Requirements specification ✓
 - Learning the languages and tools ✓

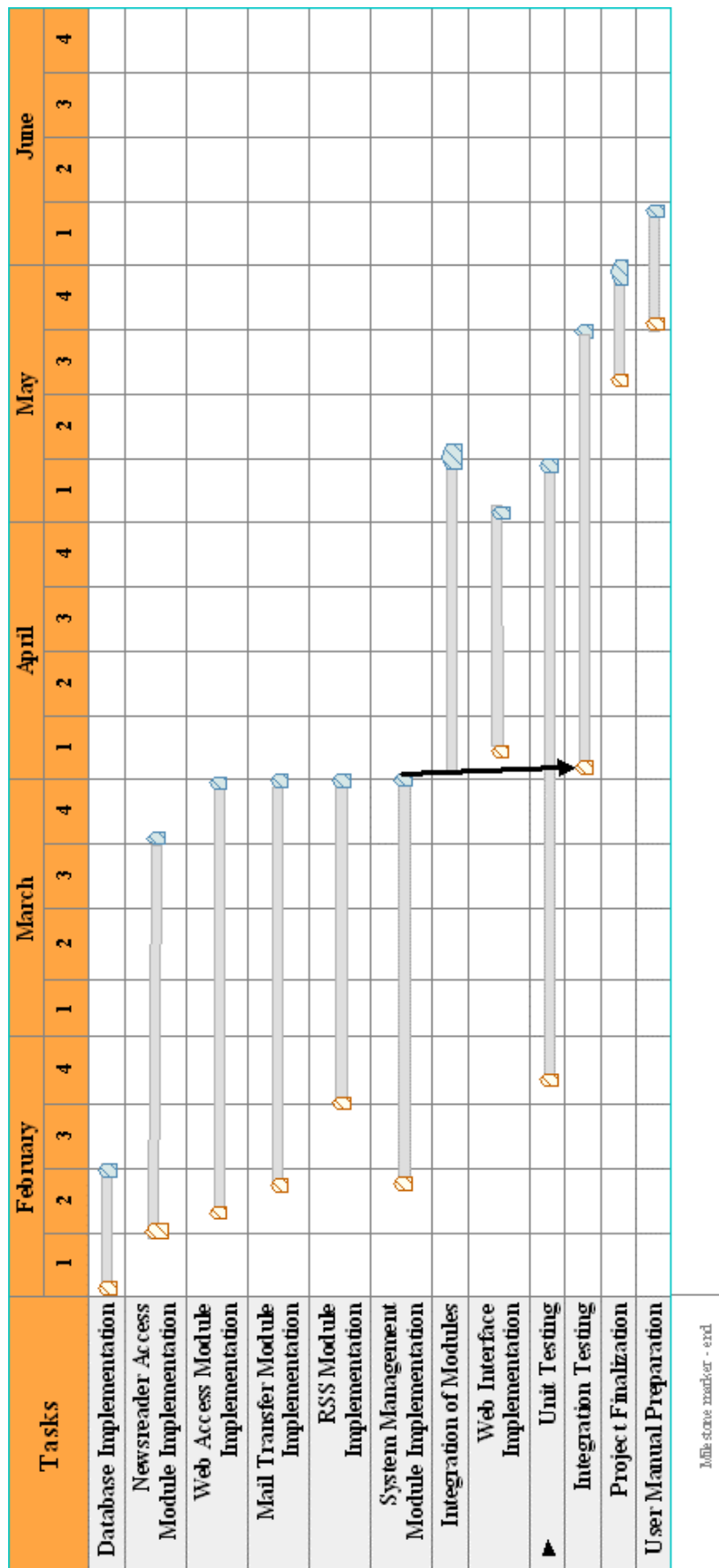
- Initial Design ✓
- Detailed Design ✓
- Implementation
- Testing
- **Documentation**
 - Project Proposal ✓
 - Analysis Report ✓
 - Initial Design Report ✓
 - Detailed Design Report ✓
 - Prototype
- **Task Set**
 - ✓ Web Access Module
 - HTTP Server construction ✓
 - Message data storage construction ✓
 - User web interface construction ✓
 - Authentication layer construction
 - Database construction
 - Read/write/reply/cancel message functionalities
 - Save/delete message functionalities
 - Search message functionality
 - File upload/download functionality
 - ✓ Newsreader Access Module
 - NNTP Server construction ✓
 - Message data storage construction ✓
 - Authentication layer construction
 - Database construction
 - Read/write/reply/cancel message functionalities
 - ✓ Mail Transfer Module
 - SMTP Server construction ✓

- Message data storage construction ✓
- Database construction
- Mail Alerting layer construction
- Send/get mail functionalities

- ✓ RSS Module
 - XML files creation ✓
 - Message data storage construction ✓
 - Database construction
 - Get RSS comments functionality
- ✓ System Management Module
 - Administrator interface construction ✓
 - Database construction
 - Basic system settings
 - Create/edit/delete user functionalities
 - Create/edit/delete newsgroup functionalities

9.2 Updated Gantt Chart

Tasks	September				October				November				December				January			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Team Organization																				
Project Topic Search																				
Proposal Report Preparation																				
Analysis of Similar Programs																				
Analysis of Open Source Servers																				
Meeting with Customer																				
Analysis of Application Areas																				
Requirement Analysis Report																				
Further Research on Software																				
Installation of all Software																				
Experimentation of all Software																				
Design of Graphical User Interface																				
Detailed Modularization of System																				
Database Design																				
Initial Design Report																				
Detailed Design of Graphical User Interfaces																				
Design of Class Hierarchy																				
Design of Modules																				
Newsreader Access Module Design																				
Web Access Module Design																				
Mail Transfer Module Design																				
RSS Module Design																				
System Management Module Design																				
Detailed Design Report																				
Prototype Development																				
Coding Prototype																				
Prototype Demo																				



10. TESTING

10.1 Unit Testing

All of the modules of our system will be tested both in a white box manner and black-box manner.

We will apply white box testing to our modules as soon as their implementation is over. We are expecting that the implementation details will not have been forgotten, so testing will be more efficient. Furthermore, we will have realized errors before they propagate to later phases of implementation, since they will require greater amount of configuration management afterwards. As our white box testing strategy, we will try to monitor the module during execution and check if the module runs as expected.

Black box testing will be used as a complementary strategy since it is useful in discovering unexpected behavior rapidly. For black-box testing, we are going to observe the input that is provided to the module and the output or the effect produced as a result. General inputs and outputs of the modules are presented below for each module.

➤ Web Access Module:

For this module, the inputs will be user login information, user commands (such as post/read articles), subjects to be searched and the XML file seeing option. The outputs of this module should be the requested newsgroups and articles which are taken from NNTP server, the search results and XML files generated by RSS Module. The kept log info is the effect produced as a result.

➤ Newsreader Access Module:

For this module, the inputs will be user login and general information and the posted messages to the NNTP server file system. Newsgroups that user has read/write permission to and their messages will be sent to user as an output.

➤ Mail Transfer Module:

For this module, the inputs will be the emails coming from users and the posted messages in the NNTP server file system. The outputs of this module should be the messages which were converted from coming emails. Sending emails to users will be the

effect produced as a result.

➤ *RSS Module:*

For this module, the inputs will be the articles in NNTP server file system and users' comments on RSS feeds. The outputs will be the generated XML files and messages to be kept in the file system.

➤ *System Management Module:*

All the admin configuration commands will be the input of this module. The changes in the configuration settings and in the database will be the effect produced as a result.

10.2 Integration Testing

Multiway is composed of different modules that have to interact with each other in order to form the whole system. To ensure that the whole system works correctly, modules that have been tested by unit testing will be added to the system one by one. After adding a new module, it will also be tested whether whole system still works correctly. That means it will be tested whether the added module still works correctly with the already integrated and tested ones.

By this integration testing, it will be ensured that the synchronized processing of modules is correct and the whole system works correctly.

10.3 Performance Tests

Our system may have high number of users. As a consequence, the performance issues will be considerable. Thus, by the performance tests, it is aimed to discover how the system will respond to the user needs. First, performance tests will be applied to each module separately. After the integration of the modules is completed, whole system will also be tested. By these performance tests, it is aimed to find out if extra measures are needed to enhance the performance.