ServerTheon Project
Initial Design Report

BY
TheonTech
# TABLE OF CONTENTS

1.0 Project Scope and Description ............................................. 4  
1.1 Problem Definition ......................................................... 4  
1.2 Project Scope ............................................................... 4  

2.0 Data Design ........................................................................... 5  
2.1 Database Description ............................................................ 5  
  2.1.1 Data Objects ............................................................... 5  
  2.1.2 Relationships .............................................................. 8  
2.2 Database Design ................................................................. 11  
  2.2.1 Database Table Specifications ........................................ 11  
  2.2.2 Database Table SQL’s .................................................. 17  

3.0 System Design ........................................................................ 25  
3.1 Class Diagrams .................................................................... 25  
  3.1.1 Webmail Module ......................................................... 25  
  3.1.2 WebForum Module ....................................................... 26  
  3.1.3 RSS Comment Module ................................................ 27  
  3.1.4 Admin Module ............................................................. 28  
  3.1.5 Common Classes .......................................................... 29  
  3.1.6 News Module ............................................................... 30  
  3.1.7 Mail Module ................................................................. 31  
  3.1.8 Web Module ............................................................... 32  
  3.1.9 CORE Module ............................................................ 33  
3.2 Sequence Diagrams ............................................................... 34  
  3.2.1 News Module ............................................................... 34  
  3.2.2 Mail Module ............................................................... 36  
  3.2.3 Web Module ............................................................... 38  
  3.2.4 Core Service .............................................................. 42  
3.3 Activity Diagrams ................................................................. 43  
  3.3.1 Registration ................................................................. 43  
  3.3.2 User Profile Update ..................................................... 44  
  3.3.3 News Module ............................................................. 45
1.0 Project Scope and Description

1.1 Problem Definition

Usenet is a set of protocols for generating, storing and retrieving news articles and for exchanging these articles among a readership which is potentially widely distributed. Usenet is one of the oldest computer network communications systems (established in 1980) still in widespread use.

Nowadays, the web forums and RSS newsfeeds are more commonly used for news broadcasting and discussion sessions. For the Unified News Exchange Server with NNTP, Mail, Web and RSS Project, we are expected to implement the followings:

- **The Message Exchange Core**: A basic threaded and secure message exchange service
- **The Extension Modules**: Modules providing e-mail lists, NNTP, WWW and RSS access methods.

1.2 Project Scope

Whenever a subscriber posts a message to one of the extension modules, the following tasks will be accomplished:

- The article will also be posted as mail to the subscriber and to the related mail group.
- The article will also be posted to the related newsgroup.
- The article will also be fed to the RSS clients on web.
- The article will also be seen on the web forum.
2.0 Data Design

2.1 Database Description

Data Design is the first step of the design procedure. Therefore, every other step is dependent to this step. To design our database, we create data object then we organize some relationships between these objects. We have defined our relationships by constructing ER diagrams in the requirement analysis phase. In the following sections we will look at data objects, their relationships, create tables and the data dictionary for data objects.

2.1.1 Data Objects

a. Client

The Client entity saves all information related to a user. Client has to be authenticated. Some of the attributes are optional, i.e. can be ‘null’. Entity contains a primary key which is Client ID. Client ID is auto-incremented with seed one. Attributes are:

- ClientID
- ClientName
- ClientPassword
- ClientSecretQuestion
- ClientSecretAnswer
- Name
- Surname
- Gender
- DateofBirth
- CountryID
- StateID
b. Admin

The Admin entity saves all information related to an admin. Admin have to be authenticated. All attributes are mandatory, i.e. can not be ‘null’. Entity contains a primary key which is AdminID. AdminID is auto-incremented with seed one. Attributes are:

- AdminID
- AdminName
- AdminPassword
- Name
- Surname
- Gender
- DateofBirth
- CountryID
- StateID
- ZipCode
- Telephone
- E-mail

c. Message

Message entity contains the information about messages. All attributes of the entity must be entered. However, ParentMessageID of first message will be null. Primary key is MessageID. MessageID is randomly generated by the system because of security reasons such as SQL injection. MessageSize is size of the message. Attributes are:
d. Attachment
Attachment entity contains the information about attachments. All entries must be entered. Primary key is AttachmentID. AttachmentID is auto-incremented with seed one. Attributes are:
- AttachmentID
- FilePath
- FileName

e. Group
Group entity saves all information about groups such as computer is a group and hardware and software is its subgroups. All entries must be entered. Primary key is GroupID. GroupID is auto-incremented with seed one. Attributes are:
- GroupID
- GroupName
- GroupDescription
- GroupParentID
f. Service

Service entity saves all information about services. Services are news, mail, forum and RSS. All entries must be entered. Primary key is ServiceID. ServiceID is auto-incremented with seed one. Attributes are:

- ServiceID
- ServiceName

2.1.2 Relationships

a. Send

Each client may send zero or many messages. Every message must be sent by exactly one client. Send relation attributes are:

- MessageID
- ClientID

b. Select0

Each client may select one or more services. Each service may be selected by zero or many client.

- ServiceID
- ClientID

c. Select1

Each client may select one or more groups. Each group may be selected by zero or many client.

- GroupID
- ClientID

d. Has

Every message has exactly one group. Every group may be owned by zero or many messages.
• MessageID  
• GroupID

e. Has1  
Each message has zero or many attachments. Every attachment must be owned by exactly one message.
  • MessageID  
  • AttachmentID

f. Edit0  
Every message may be edited by zero or one client. Each client may edit zero or many message.
  • MessageID  
  • ClientID

g. Delete0  
Every message may be deleted by zero or one client. Each client may delete zero or one message.
  • MessageID  
  • ClientID

h. AdSendMessage  
Each admin may send zero or many messages. Each message is sent by only one admin
  • AdminID  
  • MessageID

i. AdEditMessage  
Each admin may edit zero or many messages. Each message may be edited by zero or many admin.
- **AdminID**
- **MessageID**
- **AdEditDate**

**j. AdDeleteMessage**
Each admin may delete zero or many messages. Each message may be deleted zero or one admin.
- **AdminID**
- **MessageID**
- **AdDeleteDate**

**k. AdEditClient**
Each admin may edit zero or many client. Each client may be edited by zero or many admin.
- **AdminID**
- **ClientID**
- **AdEditDate**

**l. AdDeleteClient**
Each admin may delete zero or many client. Each client may be deleted by zero or one admin.
- **AdminID**
- **ClientID**
- **AdDeleteDate**

**m. AddGroup**
Each admin must add one or more group. Each group must be added by exactly one admin.
- **AdminID**
- **GroupID**
- **GroupCreationDate**
n. EditGroup
Each admin may edit zero or many group. Each group may be edited by zero or many admin.

- AdminID
- GroupID
- EditDate

o. DeleteGroup
Each admin may delete zero or many group. Each group may be deleted by zero or one admin.

- AdminID
- GroupID
- DeleteDate

2.2 Database Design
As a database management system we use MYSQL. We stored our data objects in tables. We created our tables using following codes.

2.2.1 Database Table Specifications

NOTE: PK = PRIMARY KEY
FK = FOREIGN KEY

a. Client

<table>
<thead>
<tr>
<th>Data</th>
<th>Type- Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientID</td>
<td>int, auto-increment, not null, (PK)</td>
<td>Number</td>
</tr>
<tr>
<td>ClientName</td>
<td>varchar(30), not null</td>
<td>Text</td>
</tr>
<tr>
<td>ClientPassword</td>
<td>varchar(10), not</td>
<td>Hidden Text</td>
</tr>
<tr>
<td>Name</td>
<td>varchar(30), not-null</td>
<td>Text</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------</td>
<td>------</td>
</tr>
<tr>
<td>Surname</td>
<td>varchar(30), not-null</td>
<td>Text</td>
</tr>
<tr>
<td>Gender</td>
<td>varchar(10), not-null</td>
<td>Text</td>
</tr>
<tr>
<td>YearofBirth</td>
<td>Date</td>
<td>Date</td>
</tr>
<tr>
<td>CountryID</td>
<td>Int (FK)</td>
<td>Text</td>
</tr>
<tr>
<td>StateID</td>
<td>Int (FK)</td>
<td>Text</td>
</tr>
<tr>
<td>ZipCode</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Telephone</td>
<td>varchar(15)</td>
<td>Text</td>
</tr>
<tr>
<td>E-mail</td>
<td>varchar(40)</td>
<td>Text</td>
</tr>
</tbody>
</table>

**b. Admin**

<table>
<thead>
<tr>
<th>Data</th>
<th>Type- Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdminID</td>
<td>int, auto-increment, not null, (PK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdminName</td>
<td>varchar(30), not null</td>
<td>Text</td>
</tr>
<tr>
<td>AdminPassword</td>
<td>varchar(10), not null</td>
<td>Hidden Text</td>
</tr>
<tr>
<td>Name</td>
<td>varchar(30), not null</td>
<td>Text</td>
</tr>
<tr>
<td>Surname</td>
<td>varchar(30), not null</td>
<td>Text</td>
</tr>
<tr>
<td>Gender</td>
<td>varchar(10), not null</td>
<td>Text</td>
</tr>
<tr>
<td>YearofBirth</td>
<td>Date</td>
<td>Date</td>
</tr>
<tr>
<td>Data</td>
<td>Type-Size</td>
<td>Format</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>CountryID</td>
<td>Int (FK)</td>
<td>Text</td>
</tr>
<tr>
<td>StateID</td>
<td>Int (FK)</td>
<td>Text</td>
</tr>
<tr>
<td>ZipCode</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Telephone</td>
<td>varchar(15)</td>
<td>Text</td>
</tr>
<tr>
<td>E-mail</td>
<td>varchar(40)</td>
<td>Text</td>
</tr>
</tbody>
</table>

c. Message

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>Int NOT NULL auto_incr , (PK)</td>
<td>Text</td>
</tr>
<tr>
<td>MessageText</td>
<td>Text</td>
<td>Text</td>
</tr>
<tr>
<td>MessageSubject</td>
<td>varchar(100)</td>
<td>Text</td>
</tr>
<tr>
<td>MessageSize</td>
<td>int</td>
<td>Number</td>
</tr>
<tr>
<td>MessageIsRead</td>
<td>Bit</td>
<td>Number</td>
</tr>
<tr>
<td>SendDate</td>
<td>DateTime</td>
<td>DateTime</td>
</tr>
<tr>
<td>ParentMessageID</td>
<td>Int</td>
<td>Text</td>
</tr>
</tbody>
</table>

d. Attachment

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttachmentID</td>
<td>Int NOT NULL auto_incr , (PK)</td>
<td>Number</td>
</tr>
<tr>
<td>FilePath</td>
<td>varchar(100)</td>
<td>Text</td>
</tr>
<tr>
<td>FileName</td>
<td>varchar(50)</td>
<td>Text</td>
</tr>
</tbody>
</table>

e. Group

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupID</td>
<td>Int NOT NULL auto_incr , (PK)</td>
<td>Number</td>
</tr>
<tr>
<td>GroupName</td>
<td>varchar(50)</td>
<td>Text</td>
</tr>
<tr>
<td>GroupDesc</td>
<td>text</td>
<td>Text</td>
</tr>
<tr>
<td>GroupParentID</td>
<td>int</td>
<td>Number</td>
</tr>
</tbody>
</table>

**f. Service**

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceID</td>
<td>Int NOT NULL auto_incr, (PK)</td>
<td>Number</td>
</tr>
<tr>
<td>ServiceName</td>
<td>varchar(20)</td>
<td>Text</td>
</tr>
</tbody>
</table>

**g. Send**

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>ClientID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
</tbody>
</table>

**h. Select0**

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>ClientID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
</tbody>
</table>

**i. Select1**

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>ClientID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
</tbody>
</table>

**j. Has**

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>GroupID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
</tbody>
</table>

**k. Has1**
<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AttachmentID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
</tbody>
</table>

1. Edit0

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>ClientID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>EditDate</td>
<td>DateTime</td>
<td>DateTime</td>
</tr>
</tbody>
</table>

m. Delete0

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>ClientID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>DeleteDate</td>
<td>DateTime</td>
<td>DateTime</td>
</tr>
</tbody>
</table>

n. AdSendMessage

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdminID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
</tbody>
</table>

o. AdEditMessage

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdminID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdEditDate</td>
<td>DateTime</td>
<td>DateTime</td>
</tr>
</tbody>
</table>
p. AdDeleteMessage

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdminID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdDeleteDate</td>
<td>DateTime</td>
<td>DateTime</td>
</tr>
</tbody>
</table>

r. AdEditClient

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdminID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdEditDate</td>
<td>DateTime</td>
<td>DateTime</td>
</tr>
</tbody>
</table>

s. AdDeleteClient

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdminID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdDeleteDate</td>
<td>DateTime</td>
<td>DateTime</td>
</tr>
</tbody>
</table>

t. AddGroup

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdminID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>GroupCreationDate</td>
<td>DateTime</td>
<td>DateTime</td>
</tr>
</tbody>
</table>

u. EditGroup

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdminID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>EditDate</td>
<td>DateTime</td>
<td>DateTime</td>
</tr>
</tbody>
</table>
v. DeleteGroup

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>AdminID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
<tr>
<td>DeleteDate</td>
<td>DateTime</td>
<td>DateTime</td>
</tr>
</tbody>
</table>

y. Country

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>CountryID</td>
<td>Int NOT NULL (PK)</td>
<td>Number</td>
</tr>
<tr>
<td>CountryName</td>
<td>Varchar(30) NOT NULL</td>
<td>Text</td>
</tr>
</tbody>
</table>

z. State

<table>
<thead>
<tr>
<th>Data</th>
<th>Type-Size</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateID</td>
<td>Int NOT NULL (PK)</td>
<td>Number</td>
</tr>
<tr>
<td>StateName</td>
<td>Varchar(30) NOT NULL</td>
<td>Text</td>
</tr>
<tr>
<td>CountryID</td>
<td>Int NOT NULL (FK)</td>
<td>Number</td>
</tr>
</tbody>
</table>

2.2.2 Database Table SQL’s

CREATE TABLE Country(
    CountryID INTEGER NOT NULL AUTO_INCREMENT,
    CountryName VARCHAR(30) NOT NULL,
    PRIMARY KEY(CountryID)
);

CREATE TABLE State(
CREATE TABLE State(
    StateID INTEGER NOT NULL AUTO_INCREMENT,
    StateName VARCHAR(30) NOT NULL,
    CountryID INTEGER NOT NULL,
    PRIMARY KEY (StateID),
    FOREIGN KEY (CountryID) REFERENCES Country (CountryID)
);

CREATE TABLE Client(
    ClientID INTEGER NOT NULL AUTO_INCREMENT,
    ClientName VARCHAR(30) NOT NULL,
    ClientPassword VARCHAR(15) NOT NULL,
    Name VARCHAR(30) NOT NULL,
    Surname VARCHAR(30) NOT NULL,
    Gender VARCHAR(10) NOT NULL,
    ClientSecretQuestion VARCHAR(50) NOT NULL,
    ClientSecretAnswer TEXT NOT NULL,
    YearofBirth DATE NOT NULL,
    CountryID INTEGER NOT NULL,
    StateID INTEGER NOT NULL,
    ZipCode INTEGER NOT NULL,
    Telephone VARCHAR(30),
    EMail VARCHAR(40),
    PRIMARY KEY (ClientID),
    FOREIGN KEY (CountryID) REFERENCES Country (CountryID),
    FOREIGN KEY (StateID) REFERENCES State (StateID)
);

CREATE TABLE Admin(
    AdminID INTEGER NOT NULL AUTO_INCREMENT,
    AdminName VARCHAR(30) NOT NULL,
    AdminPassword VARCHAR(15) NOT NULL,
CREATE TABLE Admin ( 
    AdminID VARCHAR(10) NOT NULL PRIMARY KEY, 
    Name VARCHAR(30) NOT NULL, 
    Surname VARCHAR(30) NOT NULL, 
    Gender VARCHAR(10) NOT NULL, 
    AdminSecretQuestion VARCHAR(50) NOT NULL, 
    AdminSecretAnswer TEXT NOT NULL, 
    YearofBirth DATE NOT NULL, 
    CountryID INTEGER NOT NULL, 
    StateID INTEGER NOT NULL, 
    ZipCode INTEGER NOT NULL, 
    Telephone VARCHAR(30), 
    EMail VARCHAR(40), 
    PRIMARY KEY(AdminID), 
    FOREIGN KEY (CountryID) REFERENCES Country(CountryID), 
    FOREIGN KEY (StateID) REFERENCES State(StateID) 
); 

CREATE TABLE Message ( 
    MessageID INTEGER NOT NULL AUTO_INCREMENT, 
    MessageText TEXT, 
    MessageSubject VARCHAR(100) DEFAULT ' [NONE] ', 
    MessageSize INTEGER NOT NULL, 
    MessageIsRead BIT DEFAULT 0, 
    SendDate DATETIME NOT NULL, 
    ParentMessageID INTEGER DEFAULT NULL, 
    PRIMARY KEY (MessageID) 
);

CREATE TABLE Attachment ( 
    AttachmentID INTEGER NOT NULL AUTO_INCREMENT, 
    FilePath VARCHAR(300), 
    PRIMARY_KEY (AttachmentID) 
)
CREATE TABLE Groups(
    GroupID INTEGER NOT NULL AUTO_INCREMENT,
    GroupName VARCHAR(50),
    GroupDesc TEXT,
    GroupParentID INTEGER DEFAULT NULL,
    PRIMARY KEY (GroupId)
);

CREATE TABLE Service(
    ServiceID INTEGER NOT NULL AUTO_INCREMENT,
    ServiceName VARCHAR(20),
    PRIMARY KEY (ServiceID)
);

CREATE TABLE Send(
    MessageID INTEGER NOT NULL,
    ClientID INTEGER NOT NULL,
    UNIQUE (MessageID),
    FOREIGN KEY (MessageID) REFERENCES Message (MessageID) ON DELETE CASCADE,
    FOREIGN KEY (ClientID) REFERENCES Client (ClientID)
);

CREATE TABLE Select0(
    ServiceID INTEGER NOT NULL,
    ClientID INTEGER NOT NULL,
    UNIQUE (ServiceID, ClientID),
    FOREIGN KEY (ServiceID) REFERENCES Service (ServiceID),
    PRIMARY KEY (AttachmentID)
FOREIGN KEY (ClientID) REFERENCES Client(ClientID)
);

CREATE TABLE Select1 (  
    GroupID INTEGER NOT NULL,  
    ClientID INTEGER NOT NULL,  
    UNIQUE (GroupID ,ClientID),  
    FOREIGN KEY (GroupID) REFERENCES Groups(GroupID ),  
    FOREIGN KEY (ClientID) REFERENCES Client(ClientID)  
);

CREATE TABLE Has (  
    MessageID INTEGER NOT NULL,  
    GroupID INTEGER NOT NULL,  
    UNIQUE (MessageID,GroupID),  
    FOREIGN KEY (MessageID) REFERENCES Message(MessageID)  
    ON DELETE CASCADE,  
    FOREIGN KEY (GroupID) REFERENCES Groups(GroupID)  
);

CREATE TABLE Has1 (  
    MessageID INTEGER NOT NULL,  
    AttachmentID INTEGER NOT NULL,  
    UNIQUE (AttachmentID),  
    FOREIGN KEY (MessageID) REFERENCES Message(MessageID)  
    ON DELETE CASCADE,  
    FOREIGN KEY (AttachmentID) REFERENCES Attachment(AttachmentID)  
);
CREATE TABLE Edit0 (  
    MessageID INTEGER NOT NULL,  
    ClientID INTEGER NOT NULL,  
    EditDate DATETIME NOT NULL,  
    UNIQUE (MessageID,EditDate),  
    FOREIGN KEY (MessageID) REFERENCES Message(MessageID)  
    ON DELETE CASCADE,  
    FOREIGN KEY (ClientID) REFERENCES Client(ClientID)  
)
;

CREATE TABLE Delete0 (  
    MessageID INTEGER NOT NULL,  
    ClientID INTEGER NOT NULL,  
    DeleteDate DATETIME NOT NULL,  
    UNIQUE (MessageID,ClientID,DeleteDate),  
    FOREIGN KEY (MessageID) REFERENCES Message(MessageID),  
    FOREIGN KEY (ClientID) REFERENCES Client(ClientID)  
)
;

CREATE TABLE AdSendMessage (  
    MessageID INTEGER NOT NULL,  
    AdminID INTEGER NOT NULL,  
    UNIQUE(MessageID),  
    FOREIGN KEY (MessageID) REFERENCES Message(MessageID)  
    ON DELETE CASCADE,  
    FOREIGN KEY (AdminID) REFERENCES Admin(AdminID)  
)
;

CREATE TABLE AdEditMessage (  
    MessageID INTEGER NOT NULL,  
    MessageID INTEGER NOT NULL,  
    AdminID INTEGER NOT NULL,  
    FOREIGN KEY (MessageID) REFERENCES Message(MessageID)  
    ON DELETE CASCADE,  
    FOREIGN KEY (AdminID) REFERENCES Admin(AdminID)  
)
;
CREATE TABLE AdEditMessage (  
  MessageID INTEGER NOT NULL,  
  AdminID INTEGER NOT NULL,  
  AdEditDate DATETIME NOT NULL,  
  UNIQUE (MessageID,AdminID,AdEditDate),  
  FOREIGN KEY (MessageID) REFERENCES Message(MessageID)  
  ON DELETE CASCADE,  
  FOREIGN KEY (AdminID) REFERENCES Admin(AdminID)  
);

CREATE TABLE AdDeleteMessage (  
  MessageID INTEGER NOT NULL,  
  AdminID INTEGER NOT NULL,  
  AdDeleteDate DATETIME NOT NULL,  
  UNIQUE (MessageID,AdDeleteDate),  
  FOREIGN KEY (MessageID) REFERENCES Message(MessageID),  
  FOREIGN KEY (AdminID) REFERENCES Admin(AdminID)  
);

CREATE TABLE AdEditClient (  
  ClientID INTEGER NOT NULL,  
  AdminID INTEGER NOT NULL,  
  AdEditDate DATETIME NOT NULL,  
  UNIQUE (ClientID,AdminID,AdEditDate),  
  FOREIGN KEY (ClientID) REFERENCES Client(ClientID)  
  ON DELETE CASCADE,  
  FOREIGN KEY (AdminID) REFERENCES Admin(AdminID)  
);

CREATE TABLE AdDeleteClient (  
  ClientID INTEGER NOT NULL,  
  AdminID INTEGER NOT NULL,  
  AdDeleteDate DATETIME NOT NULL,
CREATE TABLE AddGroup (  
   GroupID INTEGER NOT NULL,  
   AdminID INTEGER NOT NULL,  
   GroupCreationDate DATETIME NOT NULL,  
   UNIQUE (GroupID),  
   FOREIGN KEY (GroupID) REFERENCES Groups(GroupID)  
   ON DELETE CASCADE,  
   FOREIGN KEY (AdminID) REFERENCES Admin(AdminID)  
);  

CREATE TABLE EditGroup (  
   GroupID INTEGER NOT NULL,  
   AdminID INTEGER NOT NULL,  
   EditDate DATETIME NOT NULL,  
   UNIQUE (GroupID,EditDate),  
   FOREIGN KEY (GroupID) REFERENCES Groups(GroupID)  
   ON DELETE CASCADE,  
   FOREIGN KEY (AdminID) REFERENCES Admin(AdminID)  
);  

CREATE TABLE DeleteGroup (  
   GroupID INTEGER NOT NULL,  
   AdminID INTEGER NOT NULL,  
   DeleteDate DATETIME NOT NULL,  
   UNIQUE (GroupID),  
   FOREIGN KEY (GroupID) REFERENCES Groups(GroupID),  
   FOREIGN KEY (AdminID) REFERENCES Admin(AdminID));
3.0 System Design
3.1 Class Diagrams
3.1.1 Webmail Module

![Class Diagram](webmail-class-diagram.png)
3.1.2 WebForum Module
3.1.3 RSS Comment Module

RSSCommentModule

- sessionID : String
+ displayLoginForm()
- requestAuthentication(uname: String, pass: String) : boolean
+ displayCommentPage()
+ displayAuthFailedPage()
+ startSession()
- setCookie()
- readCookie()
+ endSession()

RSSCommentPage

+ displayRSSCommentForm()
- requestSendComment(header: String, comment: String) : boolean
3.1.4 Admin Module

AdminModule

- sessionID : String
+ displayLoginForm()
- requestAuthentication(uname: String, pass: String) : boolean
+ displayAdminMainPage()
+ displayAuthFailedPage()
+ startSession()
- setCookie()
- readCookie()
+ endSession()

AdminMainPage

+ displayMessageOperationsPage()
+ displayClientOperationsPage()
+ displayGroupOperationsPage()

AdminMessageOperationsPage

- requestGroupNames() : String[*]
- requestMessageHeaders(groupName: String) : String[*]
+ displayGroupSelectionForm()
+ displayMessageSelectionForm()
+ displayPostMessagePage(selectedGroup: String)
+ displayReadMessagePage(messageID: String)
+ showDeleteConfirmationDialog() : boolean
- requestDeletion(messageID: String) : boolean

AdminPostMessagePage

+ displayPostMessageForm()
- requestPost(message: String)

AdminReadMessagePage

+ displayMessage()
- requestMessage(messageID: String) : String

AdminClientOperationsPage

- requestClientData() : String[*]
- requestClientData(groupName: String) : String[*]
+ displayGroupSelectionForm()
+ displayClientSelectionForm()
+ displayEditClientPage()
- requestDeletion(clientID: String) : boolean
+ displayEditClientPage()

AdminGroupOperationsPage

- requestGroupNames() : String[*]
+ displayGroupSelectionForm()
+ showDeleteConfirmationDialog() : boolean
- requestDeletion(groupName: String) : boolean
+ showAddGroupDialog() : String
- requestAddition(groupName: String) : boolean
+ showEditGroupDialog(groupName: String) : String
- requestNameChange(oldName: String, newName: String) : boolean

AdminEditClientPage

+ displayEditClientForm()
- requestUpdate(clientId: String, data: String) : boolean
3.1.5 Common Classes

### Authentication

- `authenticateIP(username: String, password: String) : String`
- `authenticateSession(username: String, password: String) : String`

### Listener

- `port : int`
- `ip : String`
- `request : String`

- `setPort(port: int)`
- `start()`
- `stop()`
- `forwardPacketData()`

### DatabaseConnector

- `username : String`
- `password : String`
- `host : String`
- `port : int`
- `databaseName : String`
- `tableName : String`

- `setParameters(uname: String, pass: String, host: String, port: int)`
- `connect()`
- `selectDB(dbName: String)`
- `selectTable(tbName: String)`
- `executeQuery(query: String) : String`
- `close()`

### MessageConverter

- `convertFromCommon(toType: int, data: String) : String`
- `convertToCommon(data: String) : String`

<<uses>>
3.1.6 News Module

**NewsModule**

- requestQueue : String[*]  
- authenticatedClientList : String[*]

+ processCurrentRequest()  
+ newRequest(request: String)  
+ forwardToNewsServer(request: String)  
- generateServiceMessage(request: String) : ServiceMessage  
- sendServiceMessage(msg: ServiceMessage) : ServiceResponse  
- checkAuthentication() : boolean  
- isServiceRequired() : boolean  
- isDBRequired() : boolean  
- isConversionRequired() : int

**Listener**

- 1

**Authentication**

- 1

**DatabaseConnector**

- 0..*

**MessageConverter**

- 0..*
3.1.7 Mail Module

```java
MailModule
- requestQueue : String[*]
- authenticatedClientList : String[*]
+ processCurrentRequest()
+ newRequest(request: String)
- distributeUsingMailServer(request: String)
- generateServiceMessage()
- sendServiceMessage() : ServiceResponse
- isServiceRequired() : boolean
- isDBRequired() : boolean
- isConversionRequired() : int
- checkAuthentication() : boolean
- isMailToGroup() : boolean
```

- Listener
  1
  - requestQueue : String[*]
  - authenticatedClientList : String[*]

- Authentication
  1
  - requestQueue : String[*]
  - authenticatedClientList : String[*]

- DatabaseConnector
  0..*

- MessageConverter
  0..*
3.1.8 Web Module

WebModule
- requestQueue : String[*]
- authenticatedClientList : String[*]
+ processCurrentRequest()
+ newRequest(request: String)
- determineHandler() : String
- forwardToWebServer(request: String)
+ sendServiceMessage(message : ServiceMessage) : ServiceResponse
- checkAuthentication() : boolean

Listener 1 1 Authentication 1 1

RssHandler 0..* WebModule 0..*
+ handle(request : String)
- generateFeed()
- generateCommentPage()
- triggerWebServer(request: String)
- generateServiceMessage( request: String) : ServiceMessage
- isDBRequired() : boolean
- isConversionRequired() : boolean
- isServiceRequired() : boolean
- isCommentRequested() : boolean

ForumHandler 0..* 0..* 0..*
+ handle(request : String)
- generatePage()
- triggerWebServer(request: String)
- generateServiceMessage( request: String) : ServiceMessage
- isDBRequired() : boolean
- isConversionRequired() : boolean
- isServiceRequired() : boolean

AdminHandler 0..* 0..* 0..* 0..*
+ handle(request : String)
- generatePage()
- triggerWebServer(request: String)
- generateServiceMessage( request: String) : ServiceMessage
- isDBRequired() : boolean
- isConversionRequired() : boolean
- isServiceRequired() : boolean

DatabaseConnector 1 1 1 1

MessageConverter 1 1 1 1
3.1.9 CORE Module

ServerTheonCore

- messageQueue : ServiceMessage[*]

+ start()
+ stop()
- init()
- processNextMessage() : ServiceResponse
+ newServiceMessage(message: ServiceMessage)
- determineType(message: ServiceMessage) : int
- broadcastNewPost()
- performAdminOperation(message: String)
- performGroupOperation(message: String)
- generateResponse()

ServiceMessage

+ request : String
+ sender : Module
+ data : String

ServiceResponse

+ response : String
+ data : String

DatabaseConnector

Module (abstract)

NewsModule

MailModule

WebModule
3.2 Sequence Diagrams

3.2.1 News Module

Here, the response is sent by the news server.
setParameters(uname, pass, host, port): DatabaseConnector

isDBRequired(): DatabaseConnector

selectDB(dbName): DatabaseConnector

selectTable(tbName): DatabaseConnector

executeQuery(query): DatabaseConnector

queryResult: DatabaseConnector

isConversionRequired(): MessageConverter

convertToCommon(msg): MessageConverter

commonMsg: MessageConverter

convertFromCommon(msg): MessageConverter

newsMsg: MessageConverter

generateServiceMessage(request): NewsModule

serviceMessage: NewsModule

sendServiceMessage(serviceMessage): NewsModule

newServiceMessage(serviceMessage): NewsModule

serviceResponse: NewsModule

start: NewsModule

sendServiceMessage(): NewsModule

serviceResponse: NewsModule
3.2.2 Mail Module

Here, the mail client can be WebMailModule or a client using any e-mail application

Mail Client

Listener

MailModule

Authentication

newRequest(request)

processCurrentRequest()

authenticateSession(user, pass)

sessionData

checkAuthentication()

authResult

opt

[authResult = false]

authenticateSession(user, pass)

sessionData

isServiceRequired()

doesRequireService

opt

[doesRequireService = true]

service_mail

isMailToGroup()

response

Here, the response is sent by the mail server
isDBRequired()

(setParameters(uname, pass, host, port)
  connect()
  selectDB(dbName)
  selectTable(tbName)
  executeQuery(query)
  queryResult
  close())

isConversionRequired()

(doesRequireDB = true)

(convertToCommon(msg)
  commonMsg)

(doesRequireConv = 1)

(convertFromCommon(msg)
  mailMsg)

(doesRequireConv = -1)

(generateServiceMessage(request)
  serviceMessage)

(sendServiceMessage(serviceMessage)
  serviceResponse)

(newServiceMessage(serviceMessage)
  serviceResponse)
3.2.3 Web Module

Here, the web client can be WebForumModule, RSSCommentModule, AdminModule or a client using any RSS aggregator.

**Listener**

**WebModule**

**Authentication**

- request
- newRequest(request)
- processCurrentRequest()
- checkAuthentication()
- checkLoginResult()
- authResult
- opt
  - [authResult = false]
    - authenticateSession(user, pass)
    - sessionData
    - determineHandler(request)
    - handler
- alt
  - [handler = "rss"]
    - ref
      - handler_rss
  - [handler = "forum"]
    - ref
      - handler_forum
  - [handler = "admin"]
    - ref
      - handler_admin
  - [else]
    - forwardedToWebServer(request)

Here, the response is sent by the web server.
handler_rss

- WebModule
- RssHandler
- DatabaseConnector
- MessageConverter
- ServerTheonCore

```
handle (request)

opt

[doesRequireDB = true]
setParameters(uname, pass, host, port)
connect()
selectDB(dbName)
selectTable(tbName)
executeQuery(query)
queryResult
close()

isConversionRequired()

alt

[doesRequireConv = 1]
convertToCommon(msg)
commonMsg

[doesRequireConv = -1]
convertFromCommon(msg)
rssMsg

isServiceRequired()

opt

[doesRequireService = true]
generateServiceMessage(request)
serviceMessage
sendServiceMessage(serviceMessage)
newServiceMessage(serviceMessage)
serviceResponse

isCommentRequested()

[doesRequireCommentPage = true]
generateCommentPage()
generateFeed()

triggerWebServer(request)
```
3.2.4 Core Service

Here, Module is abstract class that implements NewsModule, MailModule or WebModule

: Module
: ServerTheonCore
: DatabaseConnector

newServiceMessage(msg)

loop

[massageQueue not empty]

processNextMessage()

setParameters(uname, pass, host, port)

connect()

selectDB(dbName)

selectTable(tbName)

executeQuery(query)

queryResult

close()

determineType(msg)

msgType

alt

[msgType = post]

broadcastNewPost()

[msgType = admin]

performAdminOperation(msg)

opResult

[msgType = group]

performGroupOperation(msg)

opResult

generateResponse()

serviceResponse
3.3 Activity Diagrams

3.3.1 Registration

Visitor

Fill The Form

[cancel]

[save]

[invalid form]

[valid form]

User

Presonel Info and selecting services included in the form
3.3.2 User Profile Update
3.3.3 News Module
3.3.4 Web Module
3.3.5 Mail Module

Client

Login

[Login Incorrect]

[Forgot Password]

Reset Password

[Login Correct]

Mail

[unsubscribes]

[subscribes]

Group

Message

[Mark Message]

[reply Message]

[New Message]

[read message]

[Delete Message]

[Forward Message]

Logout
4.0 User Interface Design

4.1 Description of the user interface

4.1.1 Objects and Actions

Main Page:
- Visitor can sign up using the “New Registration” link.
- Client can view his/her profile via “View My Profile” link.
- Client can access a service page via corresponding service links (“TheonForum”, “TheonNews”, “TheonMail”).

New Registration Page:
- Visitor fills the required fields of the form and selects the services which he/she wants via checkboxes.
- This form will include:
  - * TheonID (TextBox)
  - * TheonPassword (TextBox)
  - E-mail Address (TextBox)
  - * Theon Secret Question (ComboBox)
    - Mother’s birthplace
    - Favorite food
    - Favorite teacher
    - Grandfather’s occupation
  - * Theon Secret Answer (TextBox)
  - * First Name (TextBox)
  - * Last Name (TextBox)
  - * Gender (RadioButton)
    - Male
    - Female
• * Birth Year (TextBox)
• * Country (ComboBox)
• * State (ComboBox)
• * Zip Code (TextBox)
• Telephone (TextBox)

Services
- TheonForum (CheckBox)
- TheonNews (CheckBox)
- TheonRSS (CheckBox)
- TheonMail (CheckBox)

Visitor has to fill starred (*) entries for the form to be valid.

If User does not fill required filled areas, system throws an error message such as "Error: Please, fill the "..." area!" or "Error: Please, fill the (*) areas" or "Error: Password Length must be ...!" or "Error: Your E-mail address is not valid!" etc.

At the bottom of the page, there is "I accept" and "Cancel" button if client presses former registration process will be ended and the client will be directed to the main page else if client presses latter one process will be killed and client will be directed to the main page.

**User Profile Login Page:**

- Client can log into user profile page using the login group.

  • TheonID (TextBox)
  • TheonPassword (TextBox)
  • Remind Me (CheckBox)
  • Sign In (Button)
• I forgot my password! (HtmlLink)
  - If user clicks “I forgot my password” link, user will be directed to the Reset Password Page.
  - If user supplies wrong info, system throws an error message like “Invalid ID or Password!”
  - If the information is valid, the client will be directed to the user profile page.

Reset Password Page:
- User enters email.
  • E-mail Address (TextBox)
  - If user presses “Continue” button, user will be directed to the Reset Password Page2, else if user presses “Cancel” button user will be directed to the User Profile Login Page.
  - If User does not give correct data or stay E-mail Address area as blank then system throws an error message “Error: E-mail Address are not valid!” or “Error: E-mail Address Area can not be null!”

Reset Password Page2:
- There will be some account info questionnaire and secret question.
  • Name (TextBox)
  • Surname (TextBox)
  • Secret Question (Label) (Read Only)
  • Secret Answer (TextBox)
  - If user presses “Continue” button, user will be directed to the Reset Password Page3, else if user presses “Cancel” button user will be directed to the User Profile Login Page.
- If user gives wrong info, system throws a message “Error: Name or Surname are not true for “..@..” email address!” or “Error: Secret Answer are not valid!”

**Reset Password Page3:**
- User will see its email address and system will ask him/her to write a new password.
  - E-mail Address (Label) (Read Only)
  - New Password (TextBox)
  - Retype Password (TextBox)

- If user presses “Continue” button, user will be directed to the User Profile Login Page, else if user presses “Cancel” button user will kill the process and be directed to the User Profile Login Page.
- If Retype Password != New Password then system throws a message “Error: Try Again. Your supplied passwords are not same!”

**User Profile Page:**
- Client can see/update his/her personal information via a form. (As we stated above (New Registration Page))
- At the bottom of the form, there is “Save” button to submit the information on the form.
- If User gives invalid data, system throws messages as I stated above.
- Client logs out via “Logout” button and directed to the Main Page.

**Forum Login Page:**
Client can log into forum module using the login group.
• TheonID (TextBox)
• TheonPassword (TextBox)
• Remind Me (CheckBox)
• Sign In (Button)
• I forgot my password! (HtmlLink)

- If user clicks “I forgot my password” link, user will be directed to the Reset Password Page.
- If user supplies wrong info, system throws an error message like “Invalid ID or Password!”
- If the information is valid, the client/admin will be directed to the forum main page.

**Forum Main Page:**
- At the top of the page, UserName or AdminName showed.
  • UserName or AdminName (Label)
- Client can access messages in any group via clicking the link of the subject of the message. After clicking, `<Groupname> Page opens.
- Admin can lock any topic via the "Lock Topic" button.
- Admin can move any topic via "Move Topic" button.
- Client can make a search using "Search" button which is on the bottom of the page. Client can search the keyword in a specific group/topic or can do general search. When user clicks search button, user will be directed to the Search Main Page.
- Client logs out via "Logout" button.

**<Groupname> Page:**
- Messages are listed in a GridView and this GridView columns are:
  • MessageTopic (link)
• NumberofMessages (label)
• FirstMessageClientID (label)
• LastMessageRepliedDate (label)
- At the bottom right of the page there are “Search” button.
  • Search (button)
  • Search Key (TextBox)
- GridView will be paged. (style: << < > >>).
- At the top left of GridView there is a NewMessage button.
When User clicks this button, user will be directed to the Create New Message Page.
- If User clicks MessageTopic link, user will be directed to the <Message Topic> Page.

**Create New Message Page:**
- User can create a new topic and this page includes:
  • Topic (TextBox) (if this page opened after “Reply” link this entry will automatically filled such as: Re: Replied Topic)
  • Message (MultiLineTextBox) (if this page opened after “Quote” link this entry will automatically filled with Parent Message) (if this page opened after “Edit” link this entry will automatically filled with Original Message)
  • Send (Button)
  • Cancel (Button)
- If User clicks “Send” button, message will send to the group and user will be directed to the <GroupName> Page.
- If User clicks “Cancel” button, message will not send to the group and user will be directed to the <GroupName> Page.

<Message Topic> Page:
- This page includes message and their replies as a GridView. This GridView’s columns are:
  - Sender
  - Message
  - MessageDate
- GridView will be paged. (style: << < > >>).
- GridView Message column’s every row has:
  - Sent datetime (label)
  - Reply (link)
  - Quote (link)
  - Delete (link)
  - Edit (link)
- When User clicks “Reply” link, user will be directed to the Create New Message Page.
- When User clicks “Quote” link, parent message will be quoted and user will be directed to the Create New Message Page.
- When User clicks “Delete” link, user deletes his/her own message and be directed to the <Message Topic> Page.
- When User clicks “Edit” link, user will be directed to the Create New Message Page.
- At the bottom right of the page there are “Search” button.
  - Search (button)
  - Search Key (TextBox)
Search Main Page:

- Search key (TextBox) (you can put AND, OR between search words)
- Search according to Sender (TextBox)
- Search according to Date (TextBox)
- Search Place (MultiLineTextBox)
- Search (Button)
- Cancel (Button)

- If User clicks the “Search” Button, search results are shown in Search Result Page.
- If User clicks the “Cancel” Button, user will be directed to the ParentPage.

Search Result Page:

- Search Results are shown in a GridView according to Search parameter, GridView columns are set.

News Login Page:

Client can log into news module using the login group.

- TheonID (TextBox)
- TheonPassword (TextBox)
- Remind Me (CheckBox)
- Sign In (Button)
- I forgot my password! (HtmlLink)

- If user clicks “I forgot my password” link, user will be directed to the Reset Password Page.
- If user supplies wrong info, system throws an error message like “Invalid ID or Password!”
- If the information is valid, the client will be directed to the news main page.
**News Main Page:**

- Client can (un)subscribe to a group or groups which she/he wants to follow via “Options” button which leads to a list of groups which can be selected via checkboxes. Client can also see the description under the group name and number of messages in the groups near the group name. At the bottom of the group list, there is “Save” button which saves the client preferences. *(Option Page)*
- There is a ‘selected groups’ menu on the left of the page which includes links to the groups. Client can access messages via clicking these links. When user clicks this links, *<GroupName>* Page opens.
- Client logs out via “Logout” button.

**<GroupName> Page:**

- Header of this page Message Date, Subject and Author columns exist. Besides, number of pages will be shown.
- Every row of this page includes message links. Client reads messages in groups which he/she subscribed via clicking the link of the subject of the message. When user clicks this button, message details will shown under this page in *<MessageSubject>* Page.
- Client creates a new thread in a group which he/she subscribed via “New Thread” which is on the top of the page. When User clicks this button, Create New Message Page is seen under this page.
Create New Message Page:

- Subject (TextBox) (if this page opened after “Reply” link this entry will automatically filled such as: Re: Replied Topic)
- CC (TextBox) (if this page opened after “Reply” link this entry will automatically filled )
- ID of Sender (Label) (if this page opened after “Reply” link this entry will automatically filled )
- MessageArea (MultiLineTextBox)
- Quote (Button)
- Post (Button)

  -If user clicks “Quote” button, parent message will be quoted to the MessageArea.
  -If user clicks “Post” button, message will send to the group and user will notified via a message like “To see your message please click “Refresh” button”.

<Message Subject> Page:

- User will see message details.
  - Subject (label)
  - From (label)
  - Date (label)

  - User can reply a message with clicking the link “Reply”.
  When user clicks this page, Create New Message Page will appear instead of <Message Subject> Page.

Mail Login Page:

Client can log into mail module using the login group.

- TheonID (TextBox)
- TheonPassword (TextBox)
- Remind Me (CheckBox)
- Sign In (Button)
- I forgot my password! (HtmlLink)

- If user clicks “I forgot my password” link, user will be directed to the Reset Password Page.
- If user supplies wrong info, system throws an error message like “Invalid ID or Password!”
- If the information is valid, the client will be directed to the mail main page.

Mail Main Page:
- Client can (un)subscribe to a group or groups which she/he wants to follow via “Options” button which leads to a list of groups which can be selected via checkboxes.

(Options Page)
- There is an “Inbox” button which user can see messages coming from groups which he/she subscribed. When user clicks this button Inbox Page opens.
- There is a “Sent” button which user can see his/her own messages sent to groups which he/she subscribed. When user clicks this button Sent Page opens.
- There is a “Trash” button which user can see his/her own deleted messages. When user clicks this button Trash Page opens.
- There is a “New Mail” button which user can send messages to groups which he/she subscribed and user will be directed to the New Mail Page.
- User can mark a mail as Read or Unread via “Mark as Unread” or “Mark as Read” button after selecting messages via checkboxes.
- Client logs out via “Logout” button.
Note: We assumed that admin is a client.

**New Mail Page:**
- To (TextBox) (if this page opened after “Reply” button this entry will automatically filled) (if this page opened after “Forward” button this entry will left as blank)
- Subject (TextBox) (if this page opened after “Reply” button this entry will automatically filled such as: Re: Replied Topic) (if this page opened after “Forward” button this entry will automatically filled such as: Fwd: Forwarded Topic)
- Attach Files (Button)
- MessageArea (MultiLineTextBox)
- Send (Button)
- Cancel (Button)
- If User clicks Attach Files button, user will be directed to the Attachment Page.
- If user clicks the Send button, message will be send and user will be directed to the New Mail Page.
- If user clicks the Cancel button, message will not be send and user will be directed to the Parent page.

**Attachment Page:**
- Attach Files (Button)
- Cancel (Button)
- File1 (TextBox) Browse (Button)
- File2 (TextBox) Browse (Button)
- File3 (TextBox) Browse (Button)
• File4 (TextBox) Browse (Button)
• Attach More Files (HTMLlink)

- If user presses Browse(n) button, a pop-up window will appear and user select attached file via this pop up window and path will be written to the File(n) area.
- If user clicks Attach More Files link a new File(n) TextBox and Browse button will be created.
- If user Attach Files button, files are attached and user will directed to the New Mail Page
- If user presses Cancel button, job ends and user will directed to the parent page.

**Inbox Page:**

- This page includes a Gridview of messages with columns of Sender, Subject, Date, selection combo boxes and rows of
  
  • senderID(label)
  • message subject links (link)
  • Date (label)

- When user clicks links of message subject, user will be directed to the Message Page.
  
  • Delete (Button)
  • Mark (Button)

- Before deleting user have to select one or many messages via checkboxes. Deleted messages go to TrashBox.
- Before marking user have to select one or many messages via checkboxes. UnRead messages text style are bold than Read messages.

**Sent Page:**

- Format is same as Inbox Page.
Trash Page:
- Format is same as Inbox Page.

Message Page:
- Delete (Button)
- Reply (Button)
- Forward (Button)
- Date (label) (Read Only)
- From (label) (Read Only)
- Subject (label) (ReadOnly)
- To (label) (ReadOnly)
- Attachments (GridView)

- When user clicks Delete button message is deleted and user will directed to the next message page.
- When User clicks Reply button, user will directed to the New Mail Page with quoted message.
- When User clicks Forward button, user will directed to the New Mail Page with quoted message.

Admin Login Page:
- Admin can log into system using the login group (AdminID, AdminPassword, and Remind Me).
- If admin supplies wrong info, system sends an error message like “Invalid ID or Password!”
- If the information is valid, the admin will be directed to the admin main page.
Admin Main Page:
- Admin edits the information of a client via “Edit User” button which is a link to “User Profile Page” of the client.
- Admin deletes the account of a client via “Delete User” button. When admin presses this button, a MessageBox, that asks whether he/she is sure to delete, appears.
- Admin logs out via “Logout” button.

5.0 Hardware and Software Specifications

5.1 Software Specifications

| Server Side       | - Unix / Linux  
|                   | - MySQL Server  
|                   | - Web Browser   |
| Client Side       | - Java Virtual Machine  
|                   | - Unix / Linux / Windows  
|                   | - Web Browser for forum access (optional)  
|                   | - News Reader Application for newsgroup access (optional)  
|                   | - RSS Aggregator for RSS access (optional)  
|                   | - Mail Client Application for mailing list access (optional)  |
| Developers Side   | - Eclipse Java IDE  
|                   | - MySQL DBMS      |
### 5.2 Hardware Specifications

| Server Side | - Minimum 1024 MB RAM  
| - Pentium IV Processor  
| - Minimum 10 GB Hard disk |
| Client Side | - Minimum 256 MB RAM  
| - Pentium IV Processor  
| - Minimum 5 GB Hard disk |
| Developers Side | - Minimum 512 MB RAM  
| - Pentium IV Processor  
| - Minimum 5 GB Hard disk |

### 5.3 Development Tools

We will make use of the following tools during the development period of the project:

- Java EE SDK, JSP and Apache TomCat
- Eclipse Java IDE
- MySQL Database Management System (DBMS)
- Java Virtual Environment
- Apache Web Server, James NNTP and SMTP Server
### 6.0 Gantt Chart (updated)

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
<th>Finish</th>
<th>Duration</th>
<th>% Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Teaming-Up &amp; Determining Project</td>
<td>5/25/2006</td>
<td>5/30/2006</td>
<td>6d</td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td>Topic Preferences</td>
<td>6/1/2006</td>
<td>6/6/2006</td>
<td>6d</td>
<td>100%</td>
</tr>
<tr>
<td>3</td>
<td>Project Proposal</td>
<td>6/5/2006</td>
<td>6/9/2006</td>
<td>6d</td>
<td>100%</td>
</tr>
<tr>
<td>4</td>
<td>Weekly Meeting - 1</td>
<td>6/12/2006</td>
<td>6/12/2006</td>
<td>6d</td>
<td>100%</td>
</tr>
<tr>
<td>5</td>
<td>Market Research</td>
<td>6/15/2006</td>
<td>6/15/2006</td>
<td>6d</td>
<td>100%</td>
</tr>
<tr>
<td>6</td>
<td>Project Scheduling and Tracking</td>
<td>6/14/2006</td>
<td>6/17/2006</td>
<td>4d</td>
<td>100%</td>
</tr>
<tr>
<td>7</td>
<td>Requirement Analysis</td>
<td>6/15/2006</td>
<td>6/17/2006</td>
<td>4d</td>
<td>100%</td>
</tr>
<tr>
<td>8</td>
<td>Determination of Use-Cases</td>
<td>6/18/2006</td>
<td>6/20/2006</td>
<td>7d</td>
<td>100%</td>
</tr>
<tr>
<td>9</td>
<td>Data Modeling &amp; E/R Diagrams</td>
<td>6/20/2006</td>
<td>6/24/2006</td>
<td>4d</td>
<td>100%</td>
</tr>
<tr>
<td>10</td>
<td>Functional Modeling &amp; DFDs</td>
<td>6/24/2006</td>
<td>6/28/2006</td>
<td>5d</td>
<td>100%</td>
</tr>
<tr>
<td>11</td>
<td>Behavioral Modeling &amp; State Transition Diagram</td>
<td>6/29/2006</td>
<td>7/2/2006</td>
<td>5d</td>
<td>100%</td>
</tr>
<tr>
<td>12</td>
<td>Weekly Meeting - 2</td>
<td>7/2/2006</td>
<td>7/2/2006</td>
<td>6d</td>
<td>100%</td>
</tr>
<tr>
<td>13</td>
<td>Data Dictionary</td>
<td>7/3/2006</td>
<td>7/6/2006</td>
<td>3d</td>
<td>100%</td>
</tr>
<tr>
<td>14</td>
<td>Requirement Analysis Report</td>
<td>7/7/2006</td>
<td>7/7/2006</td>
<td>6d</td>
<td>100%</td>
</tr>
<tr>
<td>15</td>
<td>Data Design</td>
<td>7/11/2006</td>
<td>7/11/2006</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>16</td>
<td>Weekly Meeting - 3</td>
<td>7/18/2006</td>
<td>7/18/2006</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>17</td>
<td>Architectural Design</td>
<td>7/12/2006</td>
<td>7/18/2006</td>
<td>7d</td>
<td>0%</td>
</tr>
<tr>
<td>18</td>
<td>Weekly Meeting - 4</td>
<td>7/16/2006</td>
<td>7/16/2006</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>19</td>
<td>User Interface Design</td>
<td>7/18/2006</td>
<td>7/26/2006</td>
<td>11d</td>
<td>0%</td>
</tr>
<tr>
<td>20</td>
<td>Weekly Meeting - 5</td>
<td>7/23/2006</td>
<td>7/23/2006</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>21</td>
<td>Component - Level Design</td>
<td>7/23/2006</td>
<td>7/28/2006</td>
<td>7d</td>
<td>0%</td>
</tr>
<tr>
<td>22</td>
<td>Weekly Meeting - 6</td>
<td>7/30/2006</td>
<td>8/30/2006</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>23</td>
<td>Testing Issues</td>
<td>8/30/2006</td>
<td>12/1/2006</td>
<td>2d</td>
<td>0%</td>
</tr>
<tr>
<td>26</td>
<td>Weekly Meeting - 7</td>
<td>12/3/2006</td>
<td>12/7/2006</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>27</td>
<td>Detailed Data Design</td>
<td>12/8/2006</td>
<td>12/14/2006</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>28</td>
<td>Detailed Architectural Design</td>
<td>12/14/2006</td>
<td>12/3/2006</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>29</td>
<td>Weekly Meeting - 8</td>
<td>12/14/2006</td>
<td>12/14/2006</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>30</td>
<td>Detailed User Interface Design</td>
<td>12/17/2006</td>
<td>12/30/2006</td>
<td>14d</td>
<td>0%</td>
</tr>
<tr>
<td>31</td>
<td>Detailed Component - Level Design</td>
<td>12/28/2006</td>
<td>1/18/2007</td>
<td>11d</td>
<td>0%</td>
</tr>
<tr>
<td>32</td>
<td>Detailed Testing Issues</td>
<td>1/8/2007</td>
<td>1/14/2007</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>33</td>
<td>Detailed Design Report</td>
<td>1/15/2007</td>
<td>1/15/2007</td>
<td>6d</td>
<td>0%</td>
</tr>
<tr>
<td>34</td>
<td>Prototype Implementation</td>
<td>1/23/2007</td>
<td>1/23/2007</td>
<td>50d</td>
<td>0%</td>
</tr>
<tr>
<td>35</td>
<td>Prototype Demo</td>
<td>1/23/2007</td>
<td>1/23/2007</td>
<td>6d</td>
<td>0%</td>
</tr>
</tbody>
</table>