CTI NELSPRUIT

JAVA UNIT 6 Project

CrossFire Forum- Project 2

Tejas Dwarkaram

2012

NS2012-0340

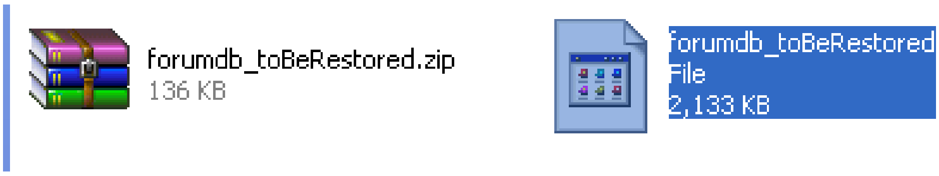
Contents

Database Restoration and Database Connection

In order to use this Web Service based Web application, your database system needs to be setup correctly.

On your resource disc, a file named “forumdb_toBeRestored.zip” shall be found. First proceed to unzip the contents.

You shall then see a file called “forumdb_toBeRestored”.

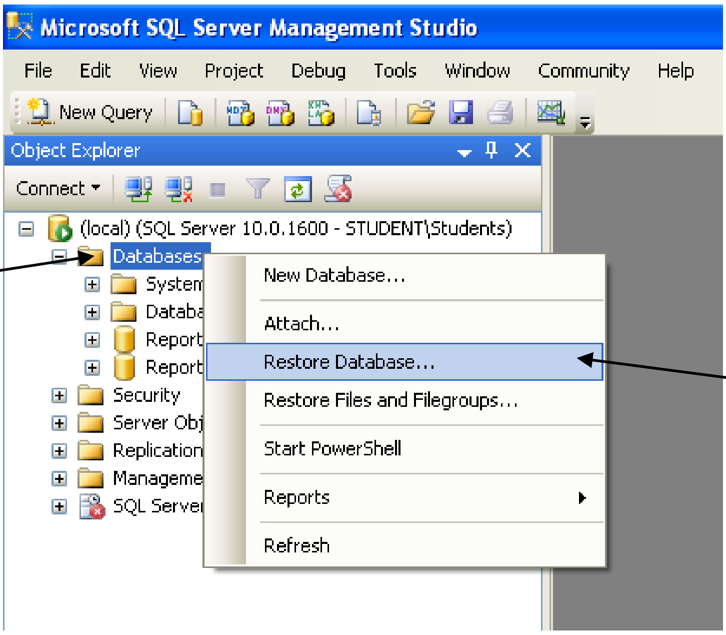


forumdb_toBeRestored.zip
136 KB

forumdb_toBeRestored
File
2,133 KB

Now proceed to MSSQL 2008.

Once opened and successfully logged in, follow the instructions below to complete the first half of setting up the database aspect of this application:

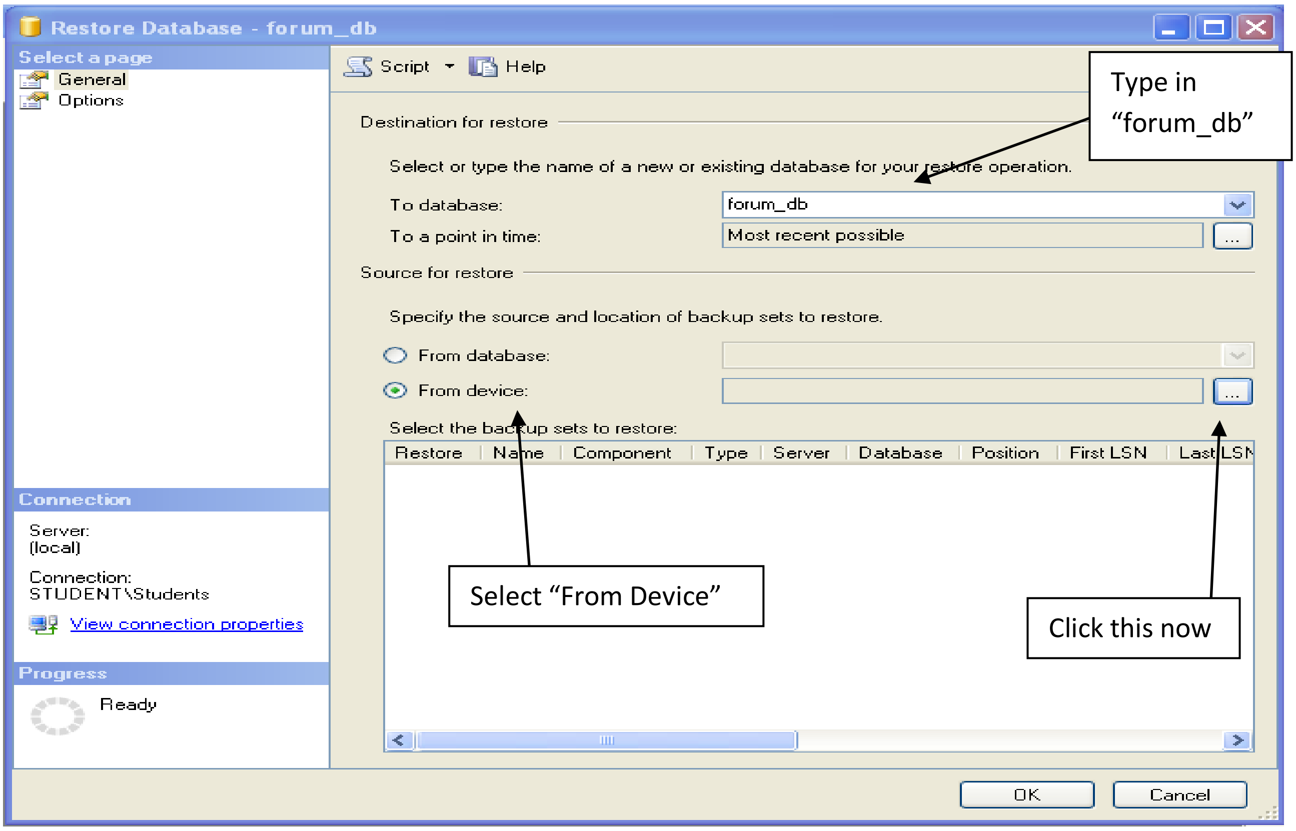


Right Click

“Databases

Now select “Restore Database ...”

There after the following screen will be shown



Restore Database - forum_db

Destination for restore

Select or type the name of a new or existing database for your restore operation.

To database: forum_db

To a point in time: Most recent possible

Source for restore

Specify the source and location of backup sets to restore.

○ From database:

● From device:

Select the backup sets to restore:

Restore	Name	Component	Type	Server	Database	Position	First LSN	Last LSN

Connection

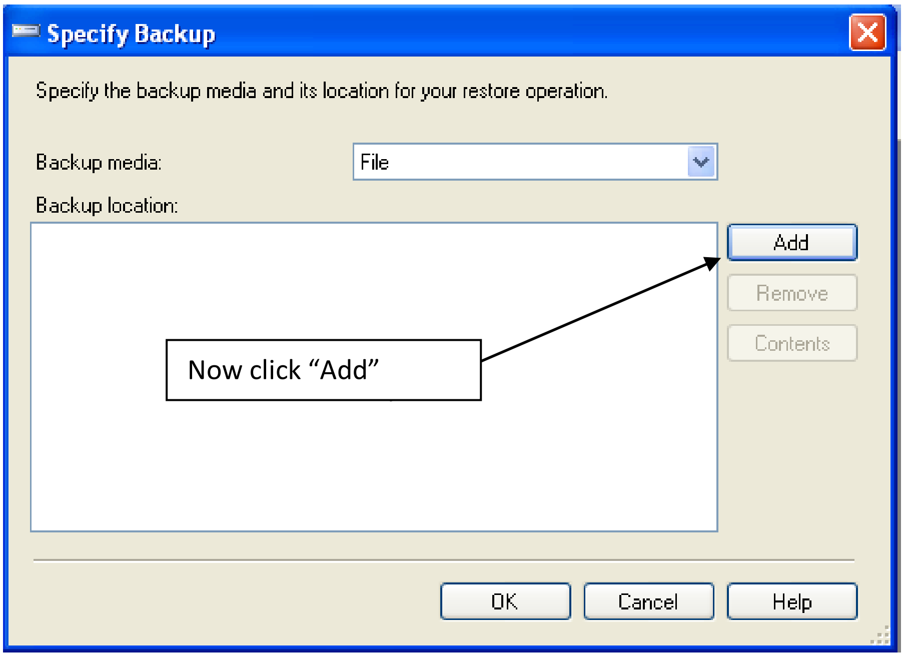
Server:
(local)

Connection:
STUDENT\Students

Type in “forum_db”

Select “From Device”

Click this now



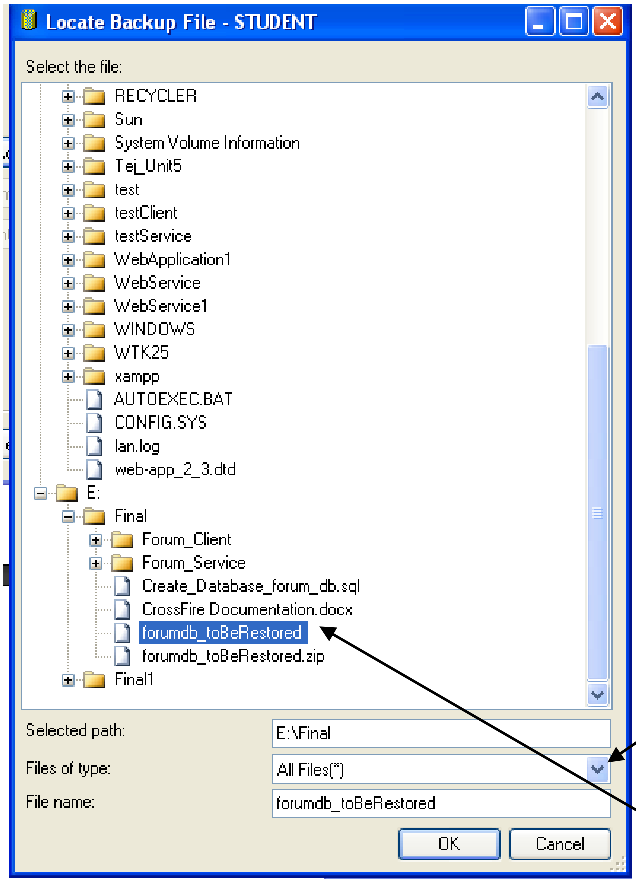
Specify Backup

Specify the backup media and its location for your restore operation.

Backup media: File

Backup location:

Now click “Add”



Locate Backup File - STUDENT

Select the file:

```
⊞ RECYCLER
⊞ Sun
⊞ System Volume Information
⊞ Tej_Unit5
⊞ test
⊞ testClient
⊞ testService
⊞ WebApplication1
⊞ WebService
⊞ WebService1
⊞ WINDOWS
⊞ WTK25
⊞ xampp
  AUTOEXEC.BAT
  CONFIG.SYS
  lan.log
  web-app_2_3.dtd
⊟ E:
  ⊟ Final
    ⊞ Forum_Client
    ⊞ Forum_Service
      Create_Database_forum_db.sql
      CrossFire Documentation.docx
      forumdb_toBeRestored
      forumdb_toBeRestored.zip
  ⊞ Final1
```

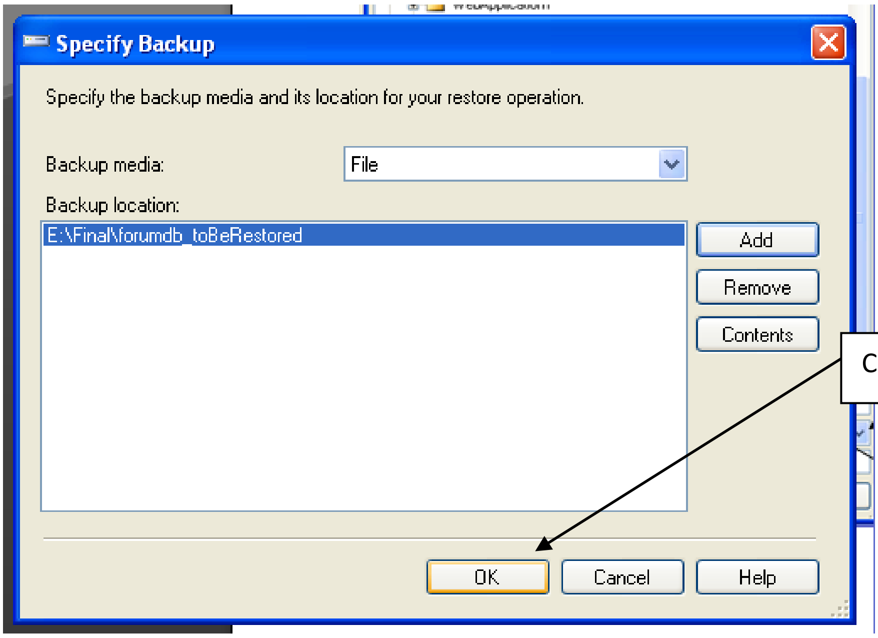
Selected path: E:\Final

Files of type: All Files(*)

File name: forumdb_toBeRestored

First Click here and select “All Files (*)”

Find the location of the file you unzipped earlier, and select it. Then click OK.



Specify Backup

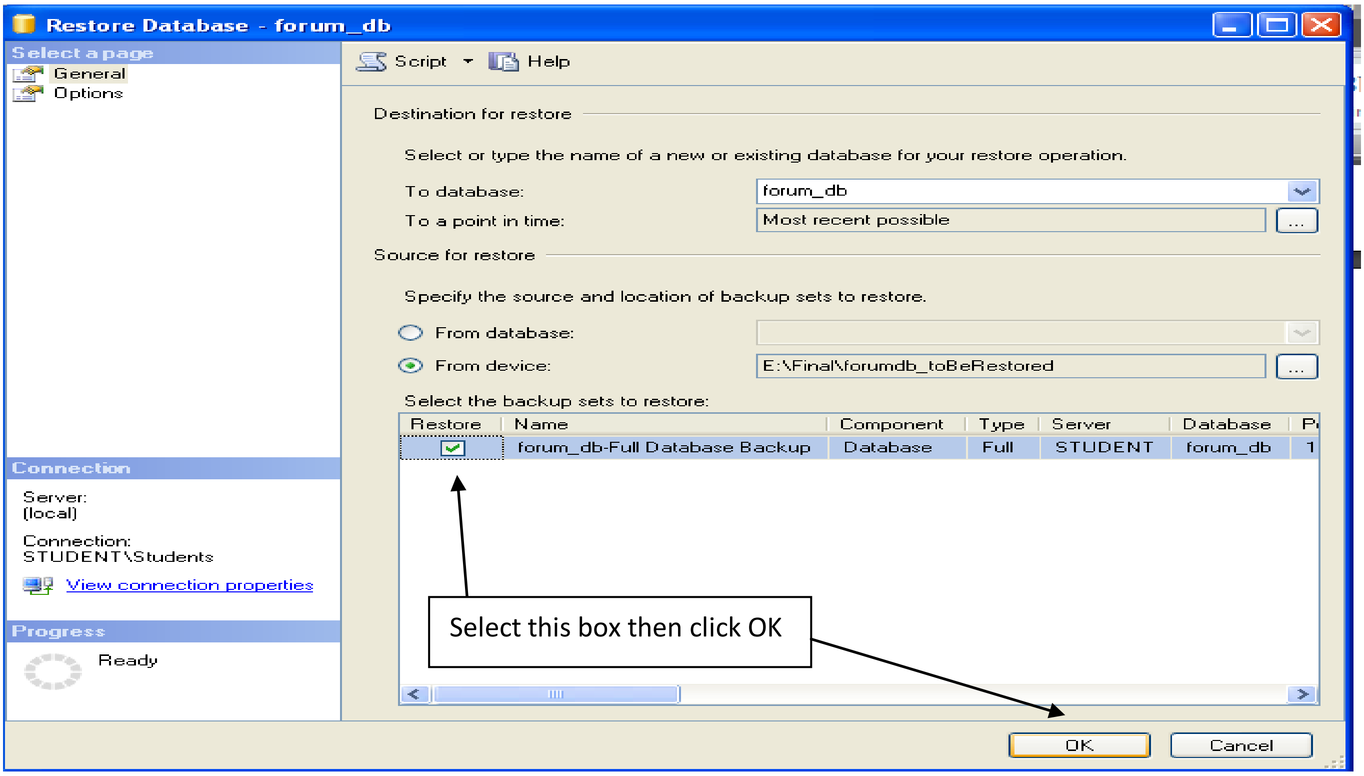
Specify the backup media and its location for your restore operation.

Backup media: File

Backup location:

E:\Final\forumdb_toBeRestored

Click OK



Destination for restore

Select or type the name of a new or existing database for your restore operation.

To database: forum_db

To a point in time: Most recent possible

Source for restore

Specify the source and location of backup sets to restore.

○ From database:

● From device: E:\Final\forumdb_toBeRestored

Select the backup sets to restore:

Restore	Name	Component	Type	Server	Database	P
☑	forum_db-Full Database Backup	Database	Full	STUDENT	forum_db	1

Connection

Server:
(local)

Connection:
STUDENT\Students

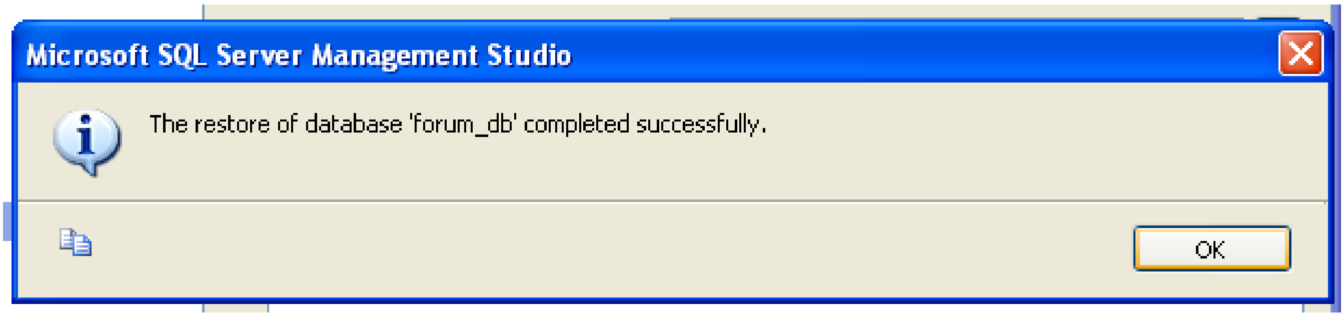
View connection properties

Progress

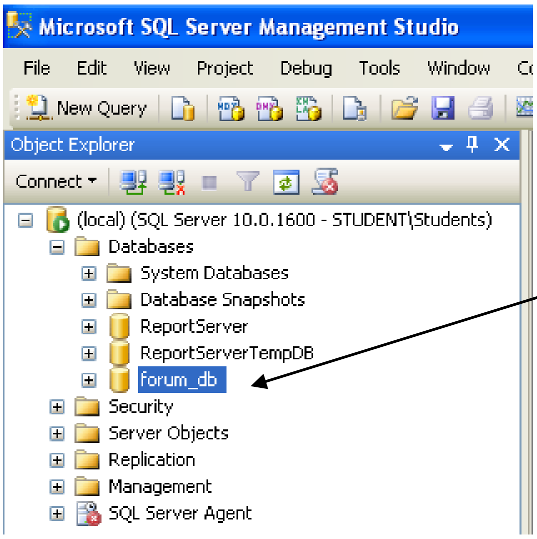
Ready

Select this box then click OK

The following message will then be shown



The restore of database 'forum_db' completed successfully.

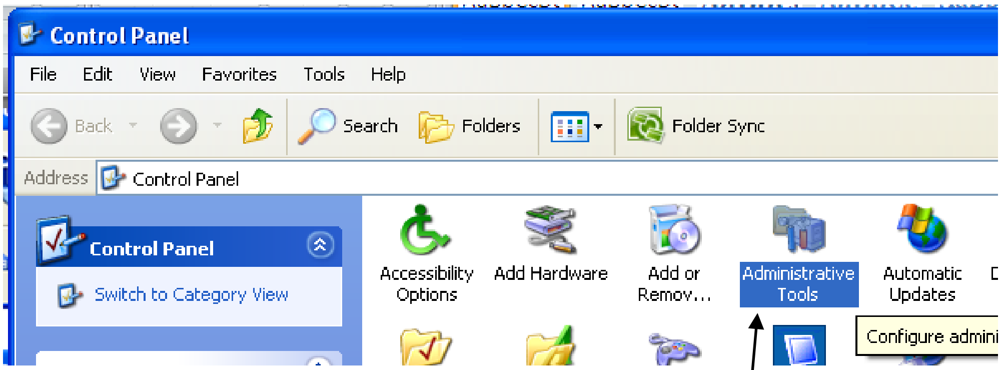


(local) (SQL Server 10.0.1600 - STUDENT\Students)
- Databases
  - System Databases
  - Database Snapshots
  - ReportServer
  - ReportServerTempDB
  - forum_db
- Security
- Server Objects
- Replication
- Management
- SQL Server Agent

You will now see “forum_db” as one of your databases

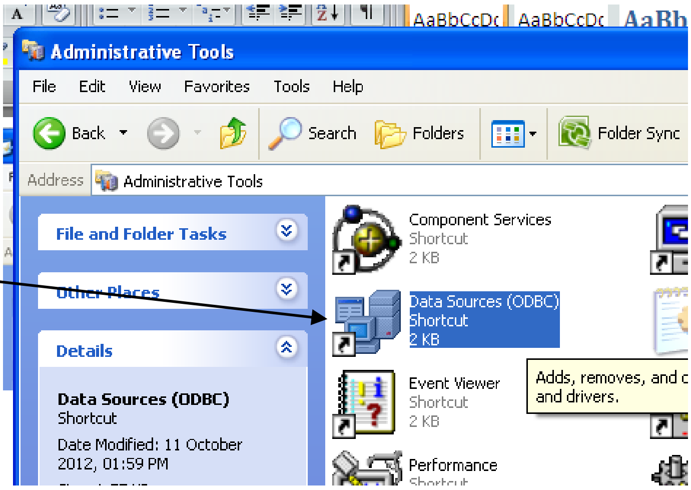
You have successfully restored the database, now to part 2.

Close MSSQL, and now go to your Control Panel of your pc, and select “Administrative Tools”



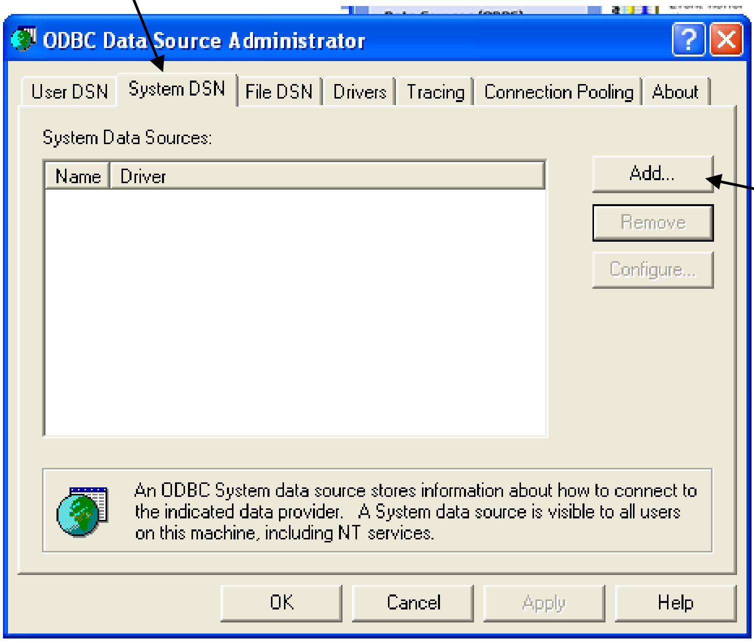
Double click on this

Now double click “Data Sources(ODBC)”


Administrative Tools
File Edit View Favorites Tools Help
Back Search Folders Folder Sync
Address Administrative Tools
File and Folder Tasks
Other Places
Details
Component Services Shortcut 2 KB
Data Sources (ODBC) Shortcut 2 KB
Event Viewer Shortcut 2 KB
Performance Shortcut
Adds, removes, and c and drivers.

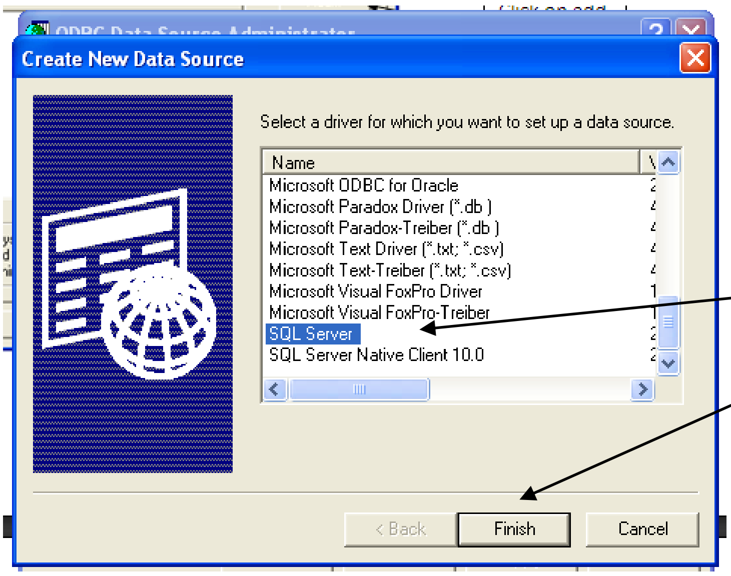
Data Sources (ODBC)
Shortcut
Date Modified: 11 October 2012, 01:59 PM

Select this tab



An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Click on add



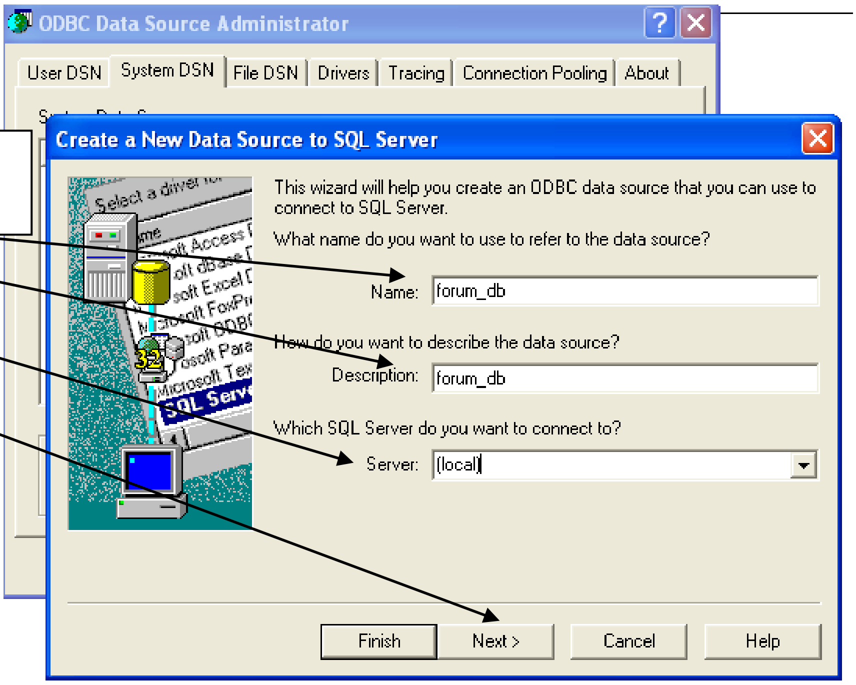
Create New Data Source

Select a driver for which you want to set up a data source.

Name
Microsoft ODBC for Oracle
Microsoft Paradox Driver (*.db)
Microsoft Paradox-Treiber (*.db)
Microsoft Text Driver (*.txt; *.csv)
Microsoft Text-Treiber (*.txt; *.csv)
Microsoft Visual FoxPro Driver
Microsoft Visual FoxPro-Treiber
SQL Server
SQL Server Native Client 10.0

Scroll down and select SQL Server then click Finish

Enter the following information and click Next



Create a New Data Source to SQL Server

This wizard will help you create an ODBC data source that you can use to connect to SQL Server.

What name do you want to use to refer to the data source?

Name: forum_db

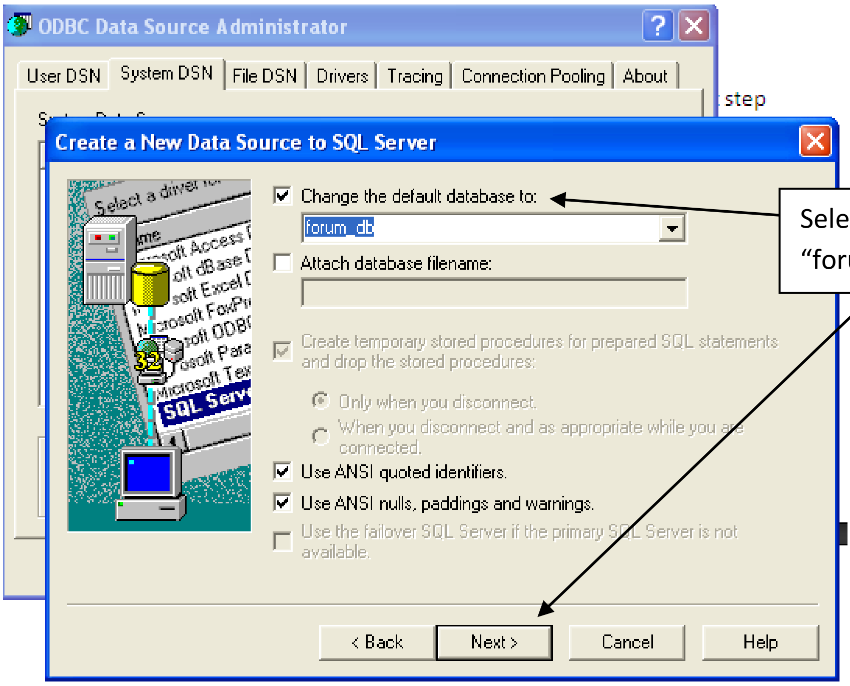
How do you want to describe the data source?

Description: forum_db

Which SQL Server do you want to connect to?

Server: (local)

Proceed to click next on the screen that follows, no information needs to be edited on that step



☑ Change the default database to:

forum_db

☐ Attach database filename:

☑ Create temporary stored procedures for prepared SQL statements and drop the stored procedures:

● Only when you disconnect.

○ When you disconnect and as appropriate while you are connected.

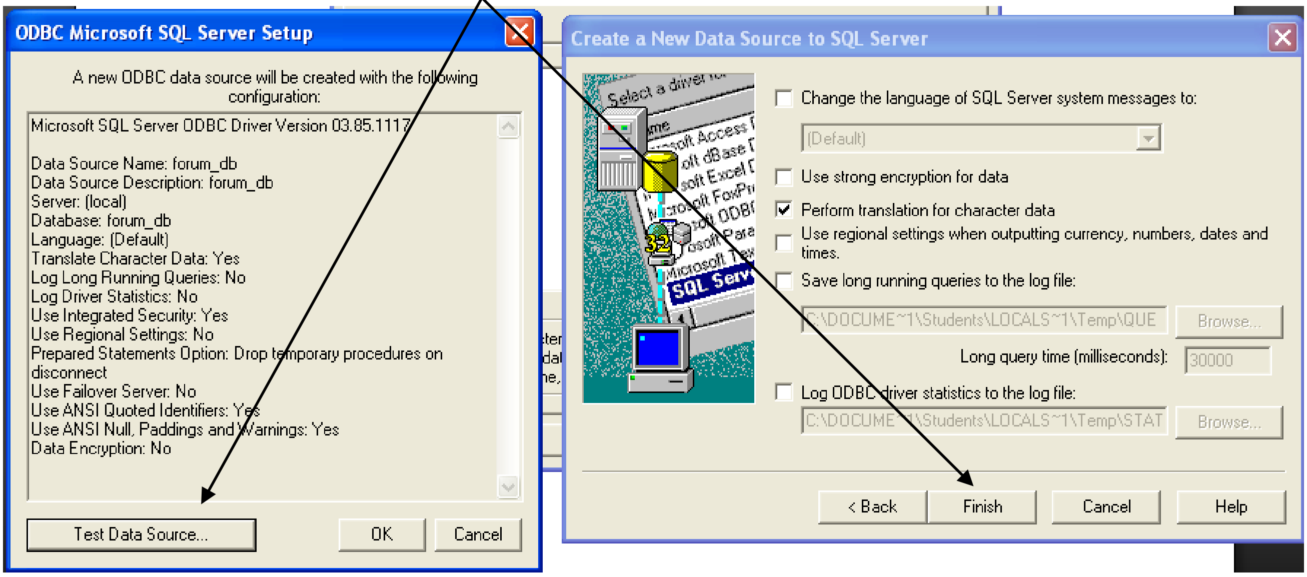
☑ Use ANSI quoted identifiers.

☑ Use ANSI nulls, paddings and warnings.

☐ Use the failover SQL Server if the primary SQL Server is not available.

Select this, and then select “forum_db” then click Next

On the next page, click finish and then select Test Data Source



A new ODBC data source will be created with the following configuration:

```
Microsoft SQL Server ODBC Driver Version 03.85.1117

Data Source Name: forum_db
Data Source Description: forum_db
Server: (local)
Database: forum_db
Language: (Default)
Translate Character Data: Yes
Log Long Running Queries: No
Log Driver Statistics: No
Use Integrated Security: Yes
Use Regional Settings: No
Prepared Statements Option: Drop temporary procedures on disconnect
Use Failover Server: No
Use ANSI Quoted Identifiers: Yes
Use ANSI Null, Paddings and Warnings: Yes
Data Encryption: No
```

☐ Change the language of SQL Server system messages to:

(Default)

☐ Use strong encryption for data

☑ Perform translation for character data

☐ Use regional settings when outputting currency, numbers, dates and times.

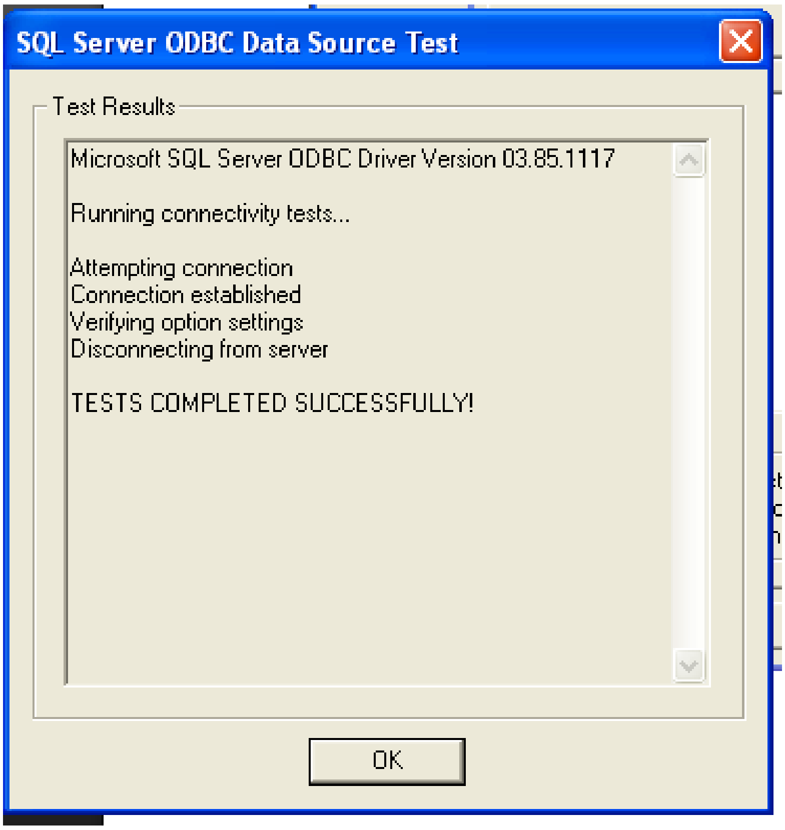
☐ Save long running queries to the log file:

C:\DOCUME~1\Students\LOCALS~1\Temp\QUE

Long query time (milliseconds): 30000

☐ Log ODBC driver statistics to the log file:

C:\DOCUME~1\Students\LOCALS~1\Temp\STAT



```
Microsoft SQL Server ODBC Driver Version 03.85.1117

Running connectivity tests...

Attempting connection
Connection established
Verifying option settings
Disconnecting from server

TESTS COMPLETED SUCCESSFULLY!
```

If everything is correct, the tests will be successful.

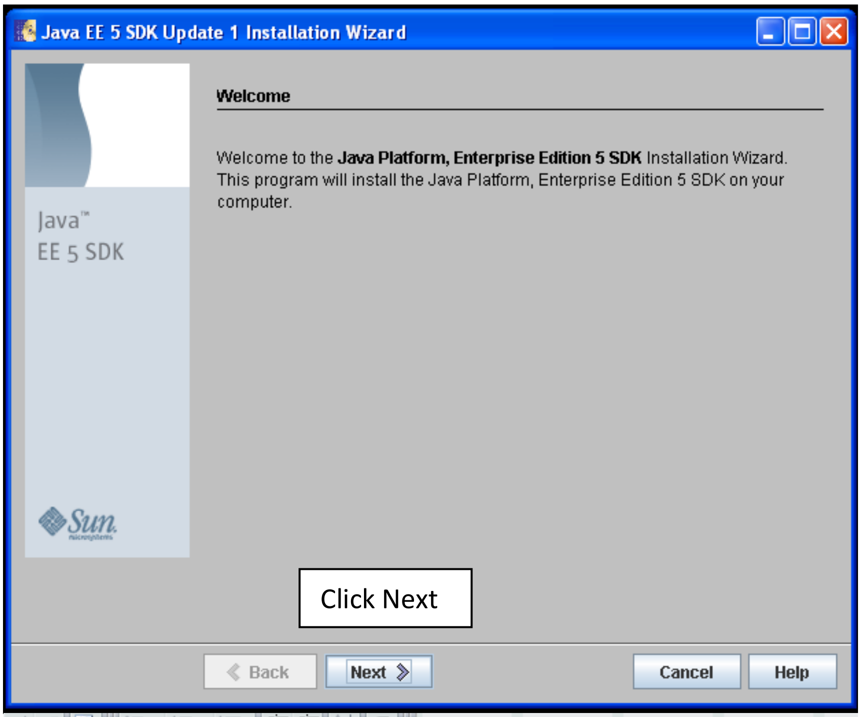
Simply click “Ok”, the “Ok” again and then “OK” again to exit.

Next I will show you how to set up the Java Sun Applications Server Program, to allow you to use the webservice correctly

Setting up the Sun Applications Server

Proceed to the resource disk again, and locate the application named "java_ee_sdk-5_01-windows-nojdk.exe", double click and run this application

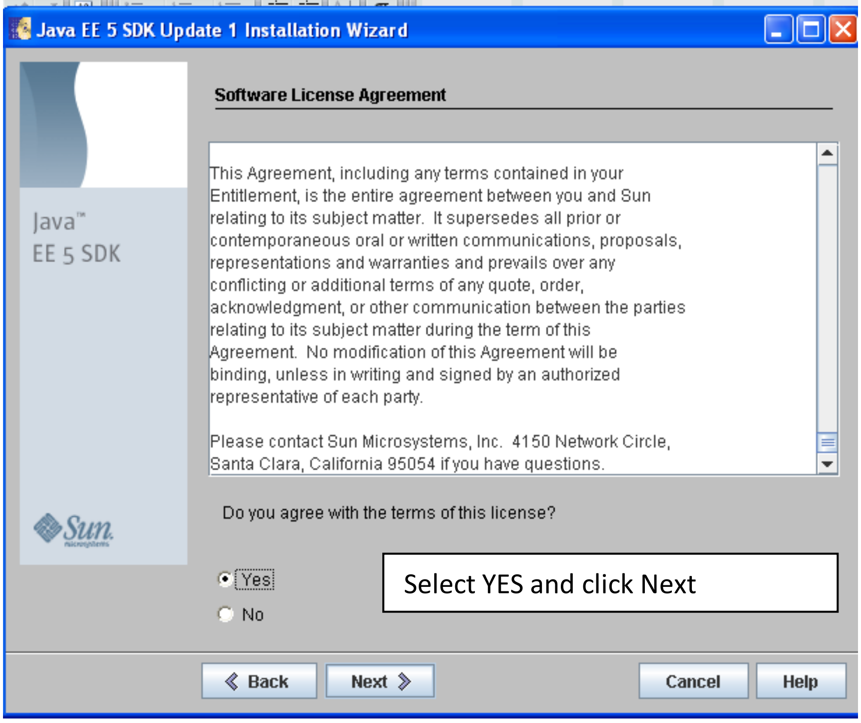
<u>NB. You require java to be properly configured on your pc to run this server program! If you don't have it properly configured, please find out how to, and then try this installation again!</u>



Welcome

Welcome to the **Java Platform, Enterprise Edition 5 SDK** Installation Wizard. This program will install the Java Platform, Enterprise Edition 5 SDK on your computer.

Click Next



Software License Agreement

This Agreement, including any terms contained in your Entitlement, is the entire agreement between you and Sun relating to its subject matter. It supersedes all prior or contemporaneous oral or written communications, proposals, representations and warranties and prevails over any conflicting or additional terms of any quote, order, acknowledgment, or other communication between the parties relating to its subject matter during the term of this Agreement. No modification of this Agreement will be binding, unless in writing and signed by an authorized representative of each party.

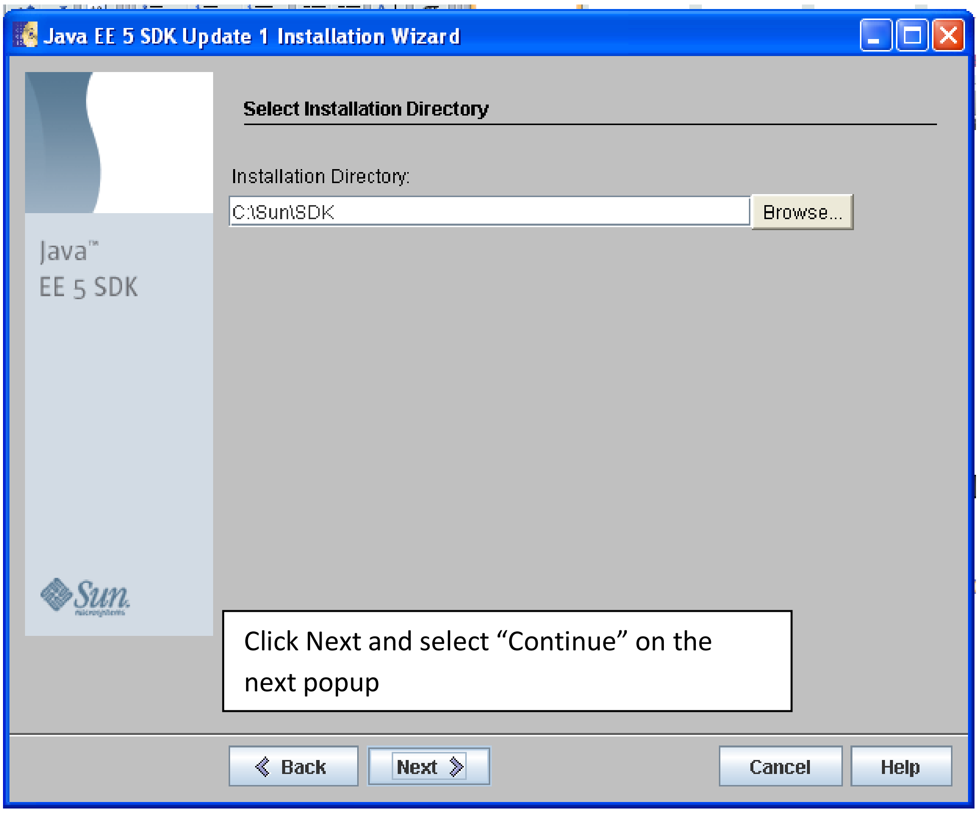
Please contact Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, California 95054 if you have questions.

Do you agree with the terms of this license?

● Yes

○ No

Select YES and click Next

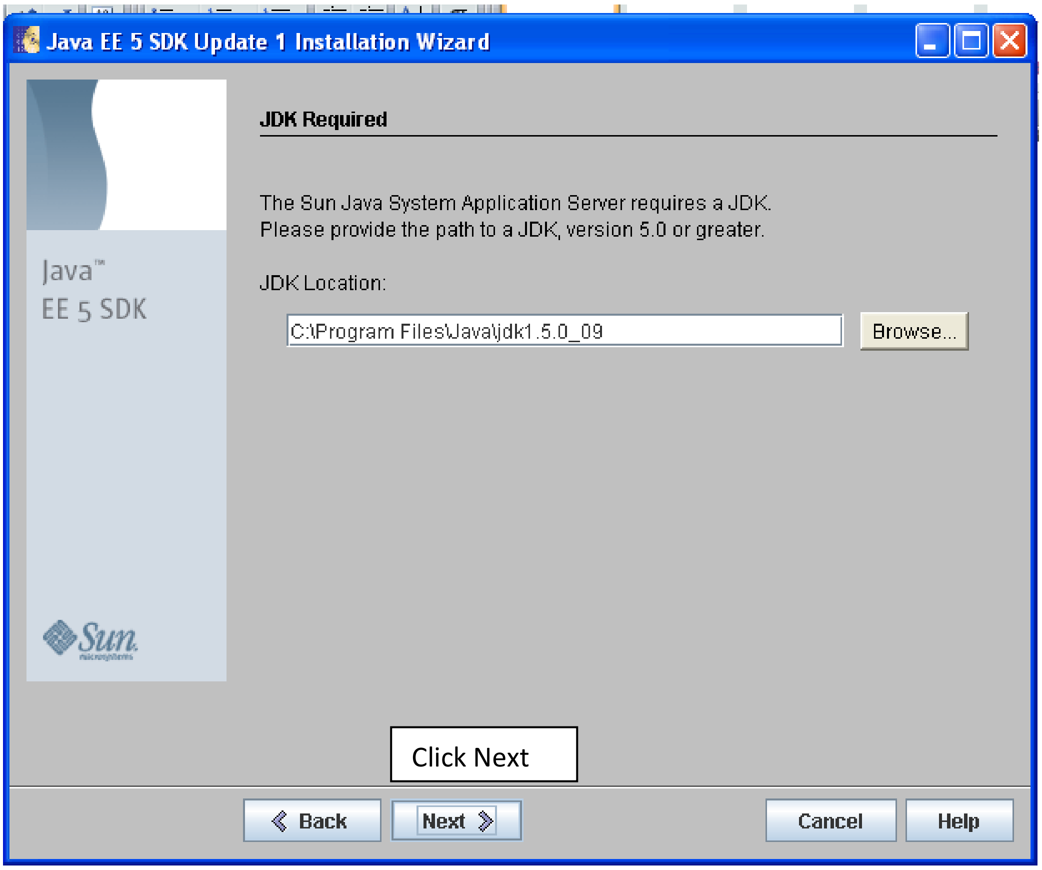


Java EE 5 SDK Update 1 Installation Wizard

Select Installation Directory

Installation Directory: C:\Sun\SDK

Click Next and select “Continue” on the next popup



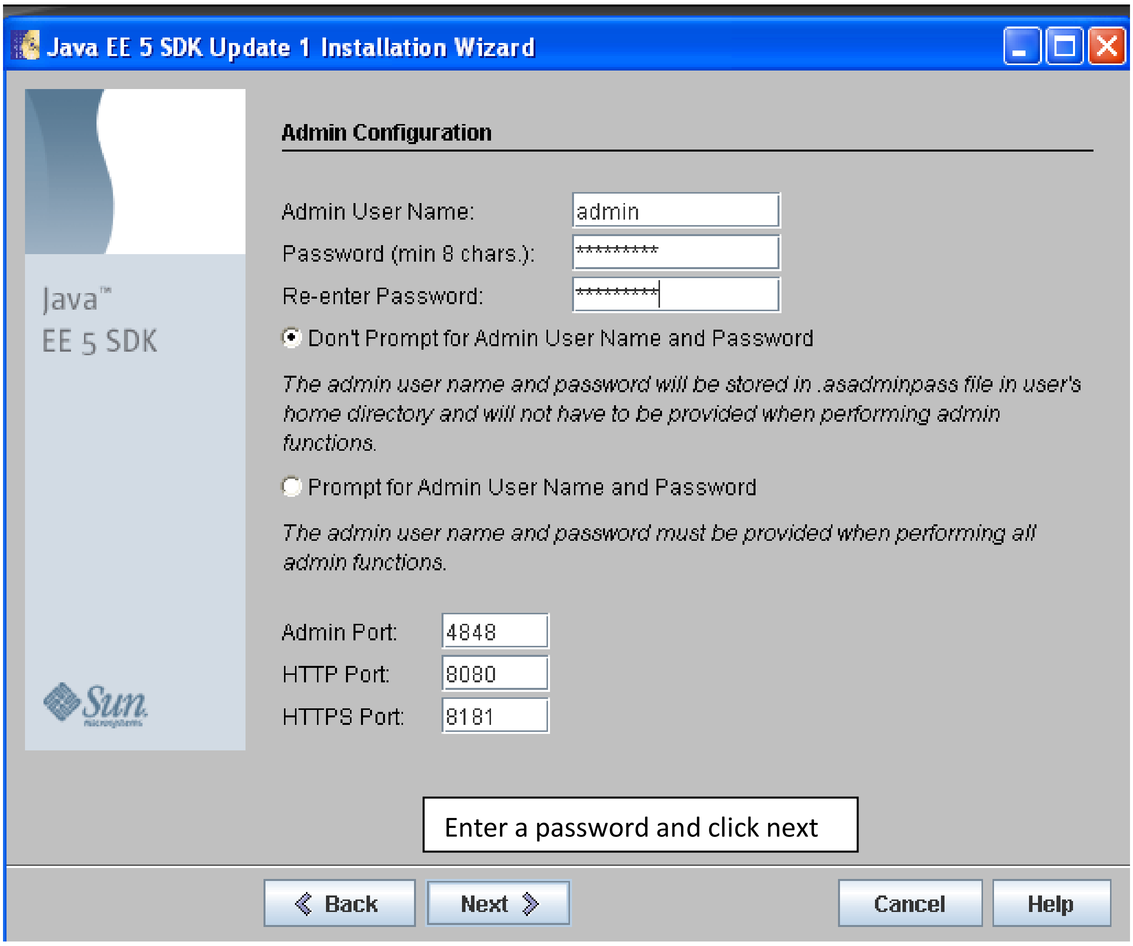
Java EE 5 SDK Update 1 Installation Wizard

JDK Required

The Sun Java System Application Server requires a JDK.
Please provide the path to a JDK, version 5.0 or greater.

JDK Location: C:\Program Files\Java\jdk1.5.0_09

Click Next



Admin Configuration

Admin User Name: admin

Password (min 8 chars.): *********

Re-enter Password: *********

● Don't Prompt for Admin User Name and Password

The admin user name and password will be stored in .asadminpass file in user's home directory and will not have to be provided when performing admin functions.

○ Prompt for Admin User Name and Password

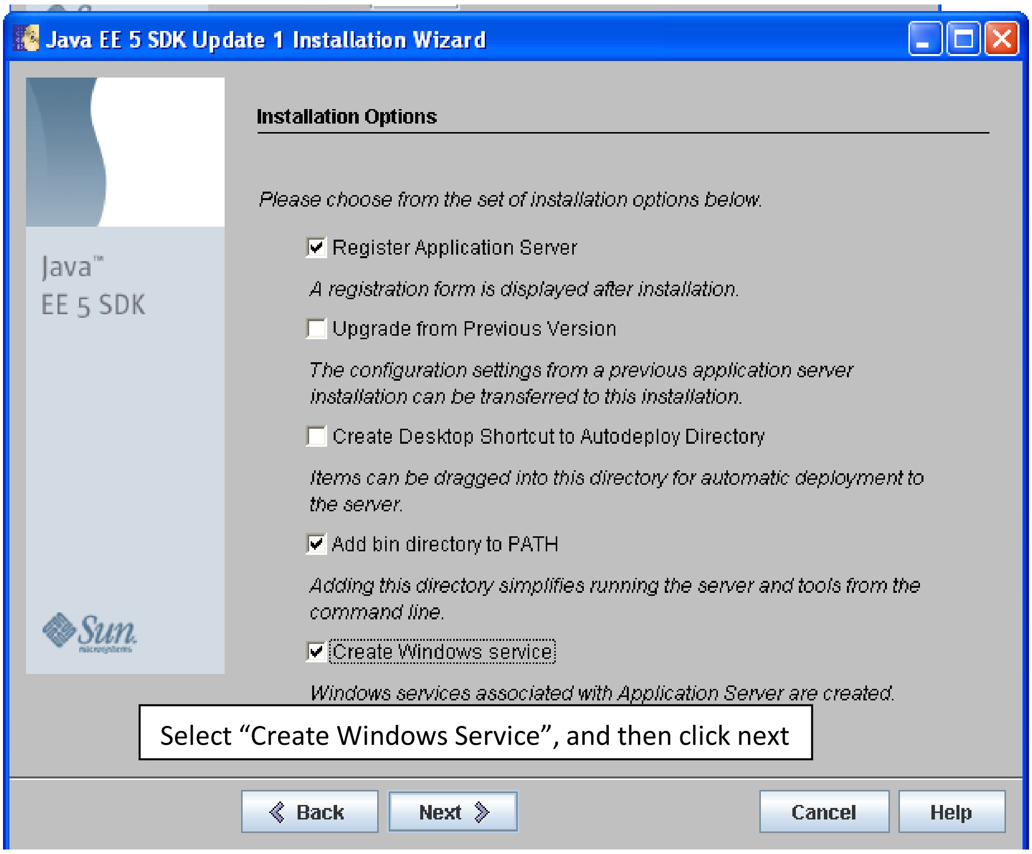
The admin user name and password must be provided when performing all admin functions.

Admin Port: 4848

HTTP Port: 8080

HTTPS Port: 8181

Enter a password and click next



Installation Options

Please choose from the set of installation options below.

☑ Register Application Server

A registration form is displayed after installation.

☐ Upgrade from Previous Version

The configuration settings from a previous application server installation can be transferred to this installation.

☐ Create Desktop Shortcut to Autodeploy Directory

Items can be dragged into this directory for automatic deployment to the server.

☑ Add bin directory to PATH

Adding this directory simplifies running the server and tools from the command line.

☑ Create Windows service

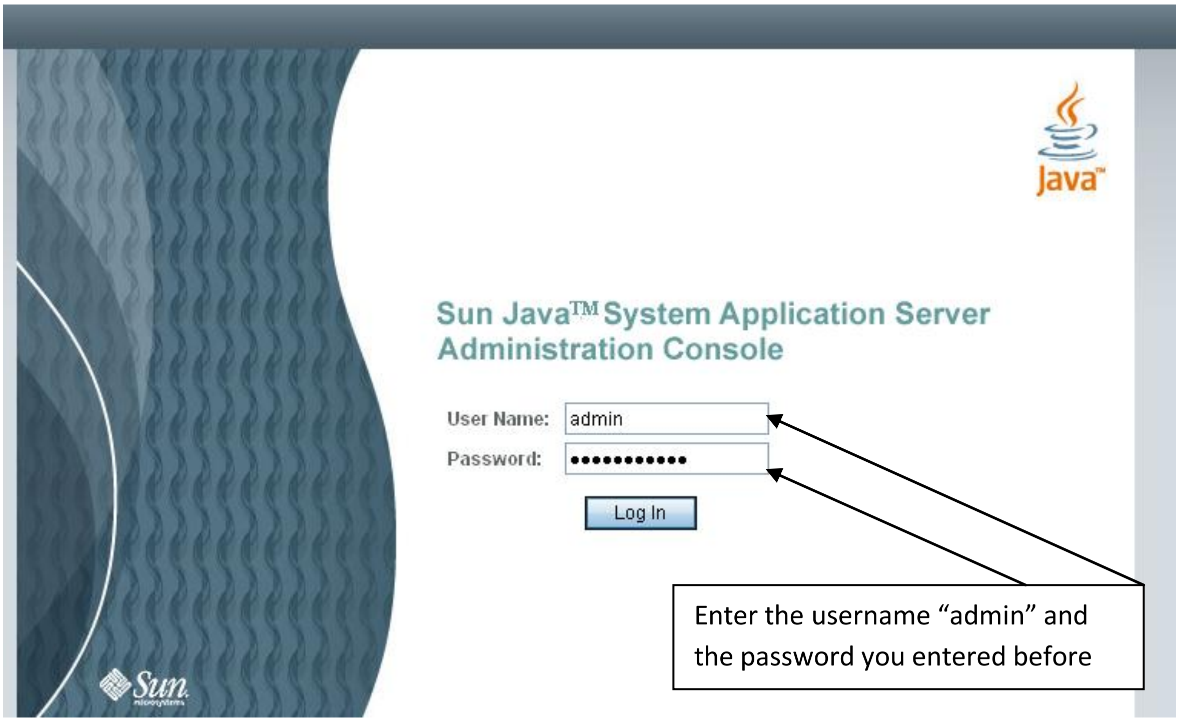
Windows services associated with Application Server are created.

Select “Create Windows Service”, and then click next

Thereafter simply press “Install Now” and wait for the installation to complete

Once complete click “Finish”, and go to your Start Menu -> Sun Microsystems -> Applications Server PE9 -> Admin Console

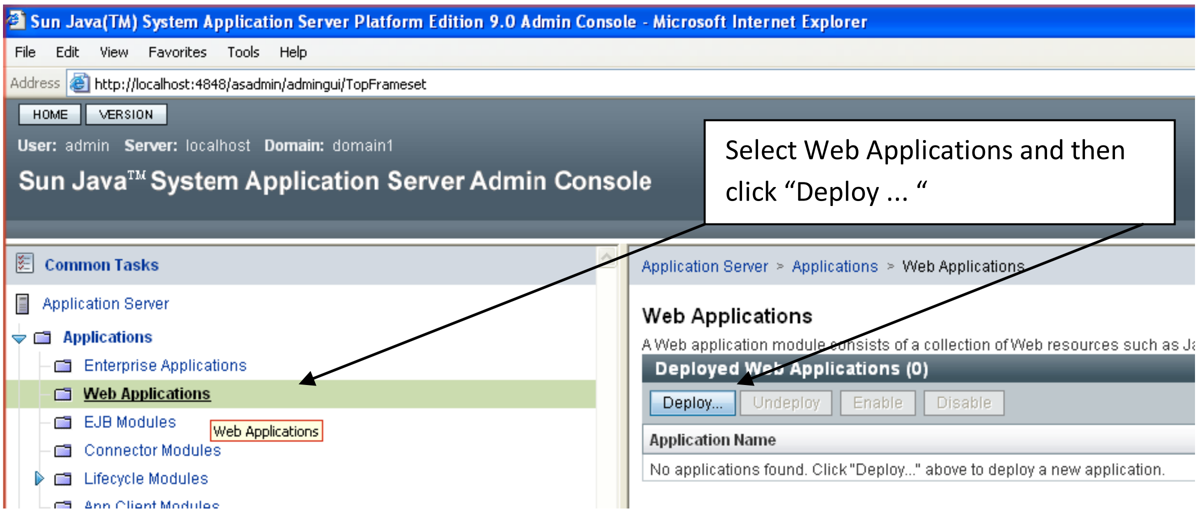
Once clicked, a web browser will open, and take you to the login page



User Name: admin

Password: ●●●●●●●●●●●●

Enter the username “admin” and the password you entered before



Sun Java(TM) System Application Server Platform Edition 9.0 Admin Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
HOME VERSION
Sun Java™ System Application Server Admin Console
Common Tasks
Application Server
Applications
Enterprise Applications
Web Applications
EJB Modules
Connector Modules
Lifecycle Modules
App Client Modules
Web Applications
Application Server > Applications > Web Applications
Deploy... Undeploy Enable Disable
Application Name

Address http://localhost:4848/asadmin/admingui/TopFrameset

User: admin Server: localhost Domain: domain1

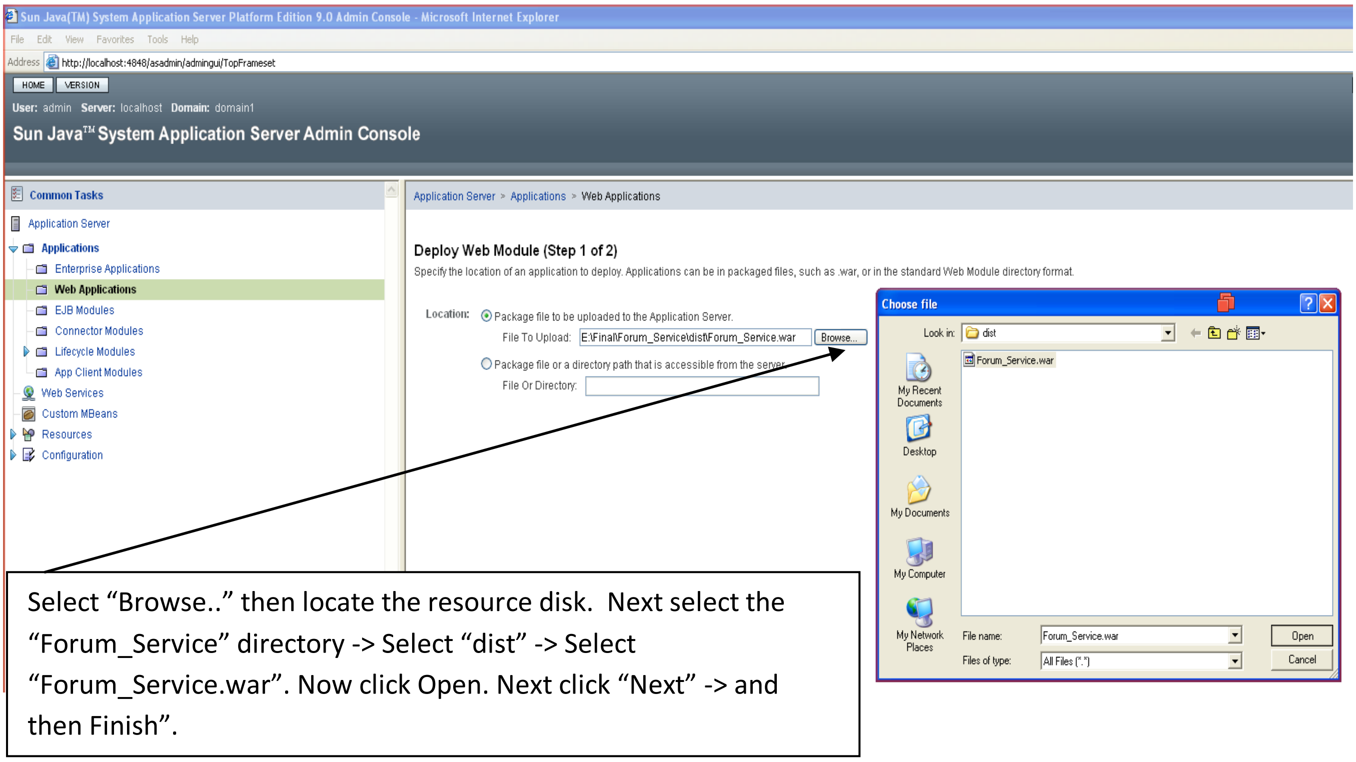
Web Applications

A Web application module consists of a collection of Web resources such as Ja

Deployed Web Applications (0)

No applications found. Click "Deploy..." above to deploy a new application.

Select Web Applications and then click “Deploy ... “



Sun Java(TM) System Application Server Platform Edition 9.0 Admin Console - Microsoft Internet Explorer

Address http://localhost:4848/asadmin/admingui/TopFrameset

User: admin Server: localhost Domain: domain1

Sun Java™ System Application Server Admin Console

Application Server > Applications > Web Applications

Deploy Web Module (Step 1 of 2)

Specify the location of an application to deploy. Applications can be in packaged files, such as .war, or in the standard Web Module directory format.

Location: ● Package file to be uploaded to the Application Server.

File To Upload: E:\Final\Forum_Service\dist\Forum_Service.war

○ Package file or a directory path that is accessible from the server.

File Or Directory:

Choose file

Look in: dist

Forum_Service.war

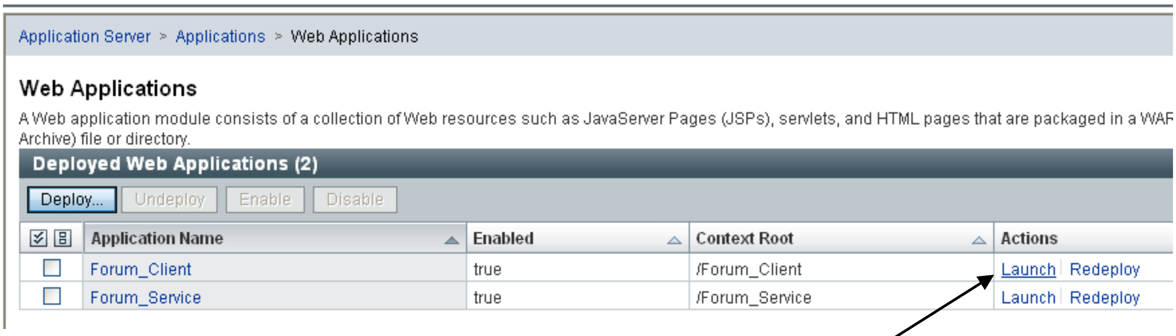
File name: Forum_Service.war

Files of type: All Files (*.*)

Select “Browse..” then locate the resource disk. Next select the “Forum_Service” directory -> Select “dist” -> Select “Forum_Service.war”. Now click Open. Next click “Next” -> and then Finish”.

Next repeat the same deployment process for the Forum_Client as well.

Once complete the following should be shown :



Application Server > Applications > Web Applications

Web Applications

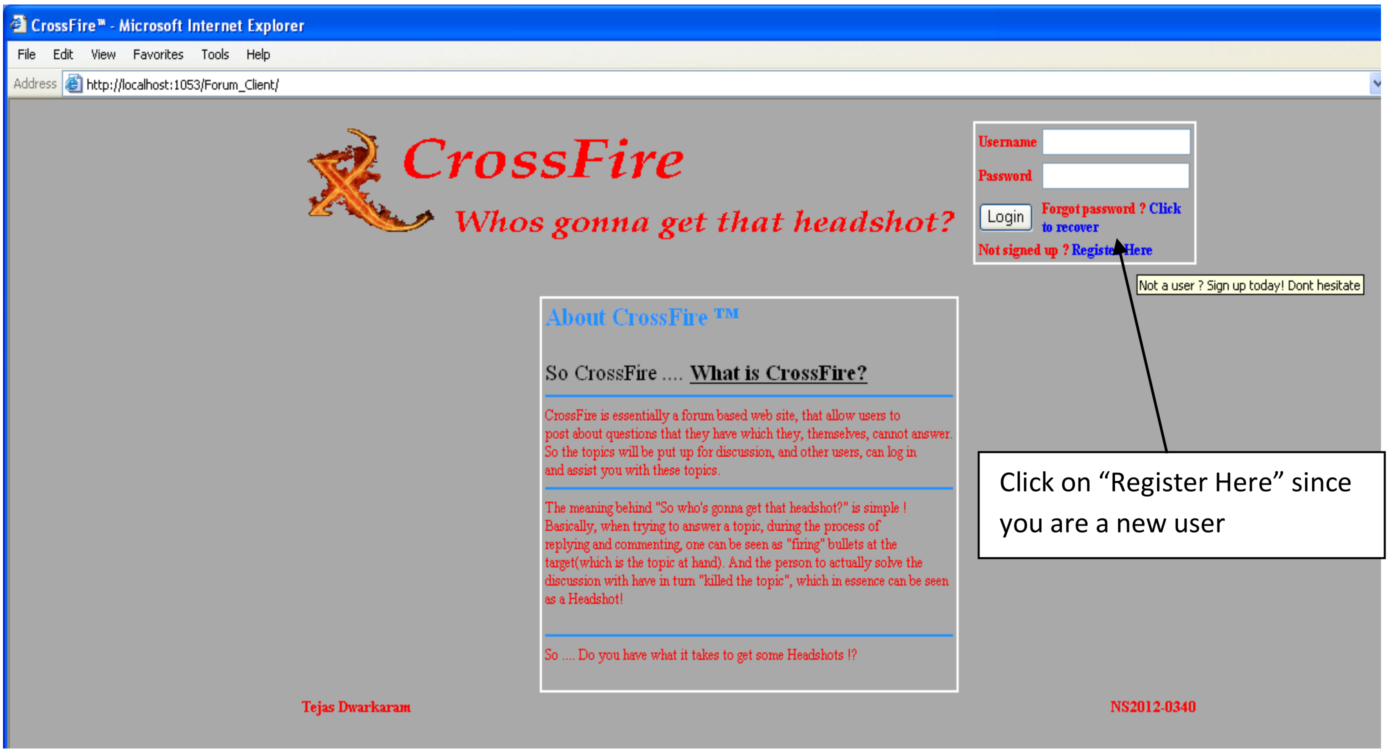
A Web application module consists of a collection of Web resources such as JavaServer Pages (JSPs), servlets, and HTML pages that are packaged in a WAR Archive) file or directory.

Deployed Web Applications (2)

☑	Application Name	Enabled	Context Root	Actions
□	Forum_Client	true	/Forum_Client	Launch Redeploy
□	Forum_Service	true	/Forum_Service	Launch Redeploy

Now click “Launch” and the index page for the website will be shown.

Using the website


CrossFire™ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:1053/Forum_Client/
CrossFire
Whos gonna get that headshot?
Username
Password
Login
Not a user ? Sign up today! Dont hesitate

Forgot password ? Click to recover

Not signed up ? Register Here

About CrossFire ™

So CrossFire .... What is CrossFire?

CrossFire is essentially a forum based web site, that allow users to post about questions that they have which they, themselves, cannot answer. So the topics will be put up for discussion, and other users, can log in and assist you with these topics.

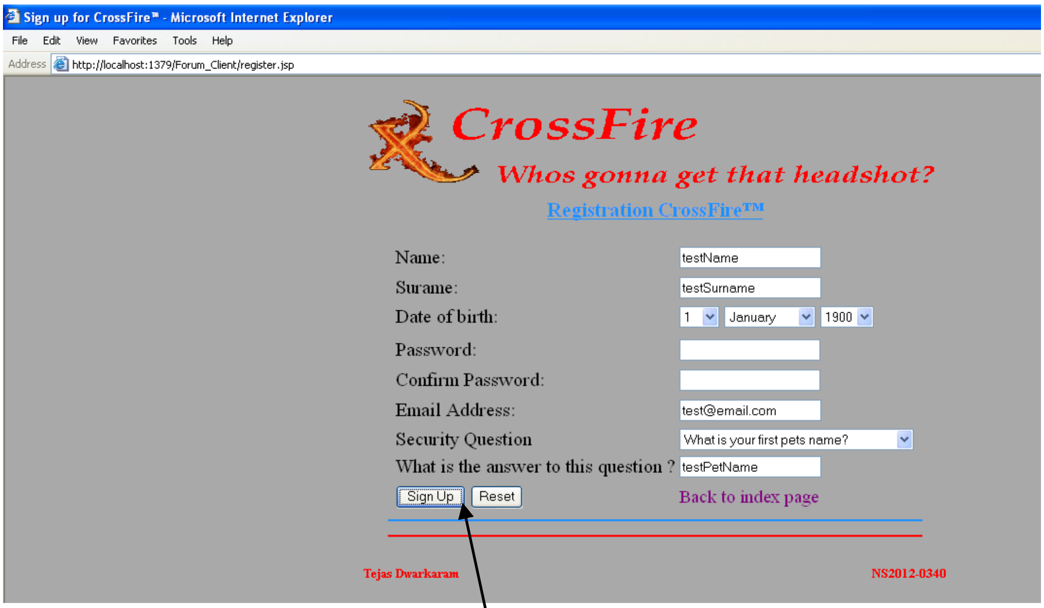
The meaning behind "So who's gonna get that headshot?" is simple ! Basically, when trying to answer a topic, during the process of replying and commenting, one can be seen as "firing" bullets at the target(which is the topic at hand). And the person to actually solve the discussion with have in turn "killed the topic", which in essence can be seen as a Headshot!

So .... Do you have what it takes to get some Headshots !?

Tejas Dwarkaram

NS2012-0340

Click on "Register Here" since you are a new user


Sign up for CrossFire™ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:1379/Forum_Client/register.jsp
CrossFire
Whos gonna get that headshot?
Sign Up
Reset

Registration CrossFire™

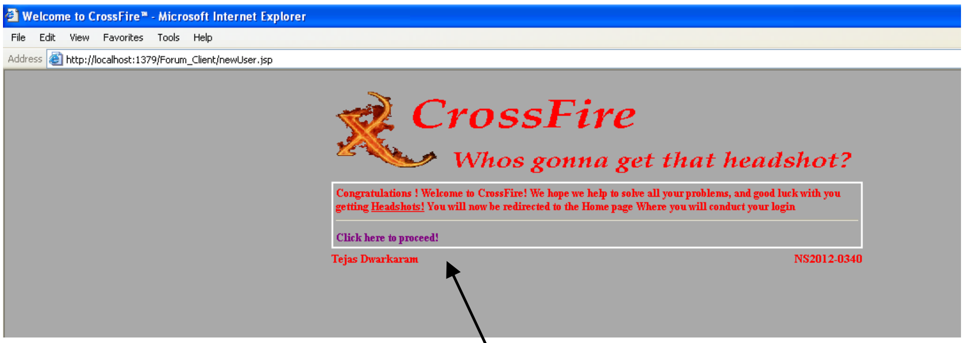
Name:	testName
Surame:	testSurname
Date of birth:	1 January 1900
Password:	
Confirm Password:	
Email Address:	test@email.com
Security Question	What is your first pets name?
What is the answer to this question ?	testPetName

Back to index page

Tejas Dwarkaram

NS2012-0340

Enter your details and click "Sign Up here". Please note : No spaces allowed, Email Address contain "@" and a ".", All passwords between 8 – 24 characters and they must match


Welcome to CrossFire™ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:1379/Forum_Client/newUser.jsp
CrossFire
Whos gonna get that headshot?

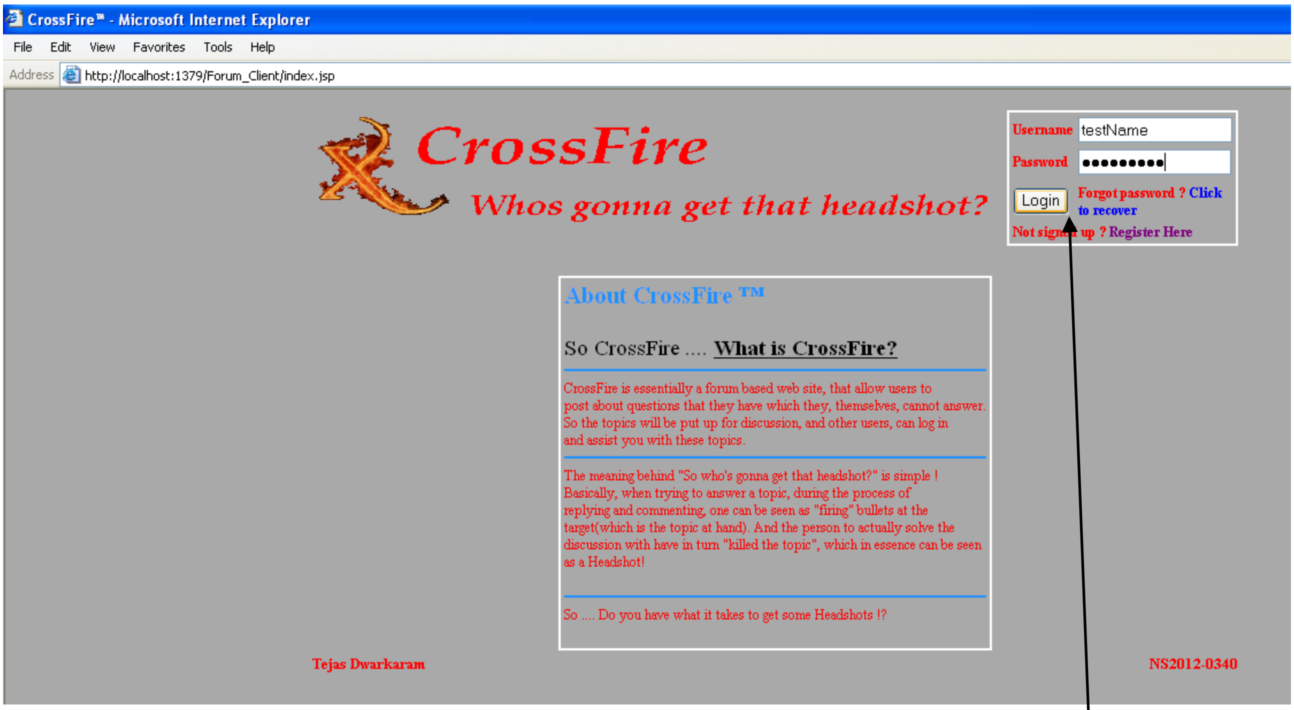
Congratulations ! Welcome to CrossFire! We hope we help to solve all your problems, and good luck with you getting Headshots! You will now be redirected to the Home page Where you will conduct your login

Click here to proceed!

Tejas Dwarkaram NS2012-0340

If the login is successful, the following page will be shown, then click “Click here to proceed, to go to the login page and login with the details you specified earlier.

If the email address is already in use, or any errors occurred with details enter, you will be notified and asked to re-enter them


CrossFire™ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:1379/Forum_Client/index.jsp
CrossFire
Whos gonna get that headshot?
Login
Forgot password ? Click to recover
Not signed up ? Register Here

Username testName

Password ●●●●●●●●●

About CrossFire ™

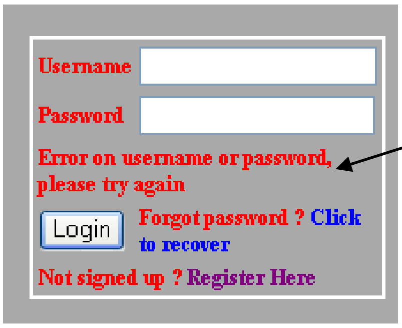
So CrossFire .... What is CrossFire?

CrossFire is essentially a forum based web site, that allow users to post about questions that they have which they, themselves, cannot answer. So the topics will be put up for discussion, and other users, can log in and assist you with these topics.

The meaning behind "So who's gonna get that headshot?" is simple ! Basically, when trying to answer a topic, during the process of replying and commenting, one can be seen as "firing" bullets at the target(which is the topic at hand). And the person to actually solve the discussion with have in turn "killed the topic", which in essence can be seen as a Headshot!

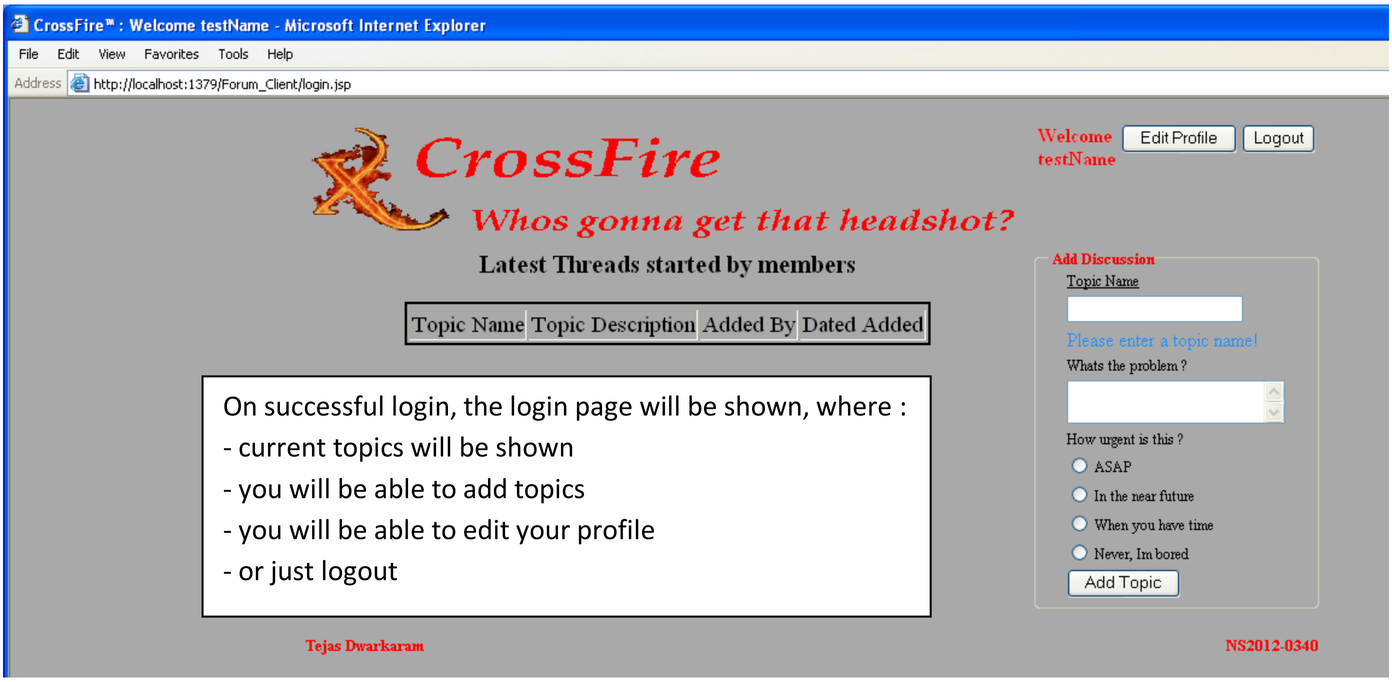
So .... Do you have what it takes to get some Headshots !?

Tejas Dwarkaram NS2012-0340


Username
Password
Login
Forgot password ? Click to recover
Not signed up ? Register Here

Error on username or password, please try again

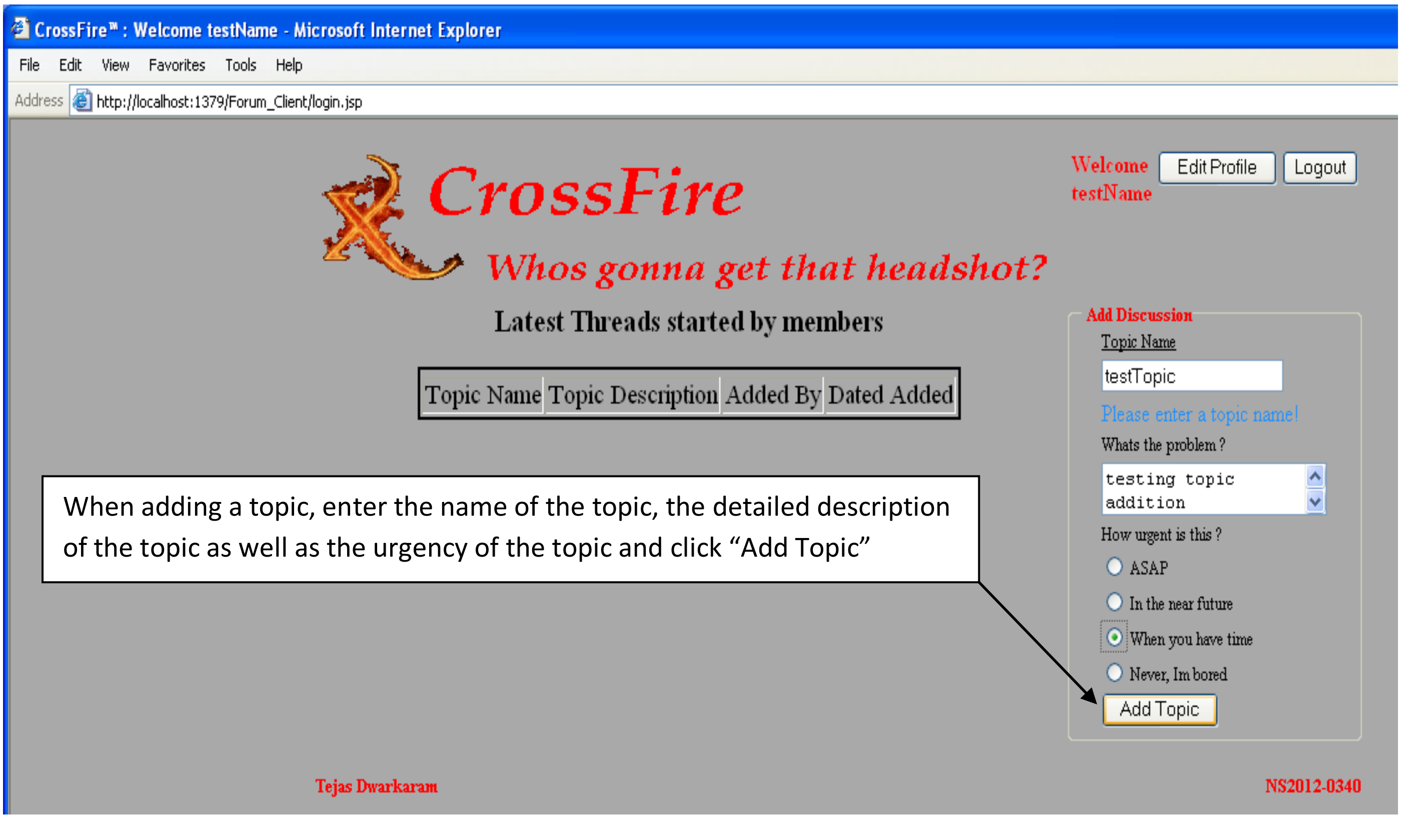
Enter the name that was used to create the account as well as the password you entered, and click login. If either are incorrect, the following will be shown, then you will be asked to re-enter.


CrossFire™ : Welcome testName - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:1379/Forum_Client/login.jsp
CrossFire
Whos gonna get that headshot?
Welcome testName
Edit Profile
Logout
Latest Threads started by members
Add Discussion
Topic Name
Please enter a topic name!
Whats the problem ?
How urgent is this ?
○ ASAP
○ In the near future
○ When you have time
○ Never, Im bored
Add Topic
Tejas Dwarkaram
NS2012-0340

Topic Name	Topic Description	Added By	Dated Added

On successful login, the login page will be shown, where :

- current topics will be shown
- you will be able to add topics
- you will be able to edit your profile
- or just logout


CrossFire™ : Welcome testName - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:1379/Forum_Client/login.jsp
CrossFire
Whos gonna get that headshot?
Welcome testName
Edit Profile
Logout
Latest Threads started by members
Add Discussion
Please enter a topic name!
Add Topic
Tejas Dwarkaram
NS2012-0340

Topic Name	Topic Description	Added By	Dated Added

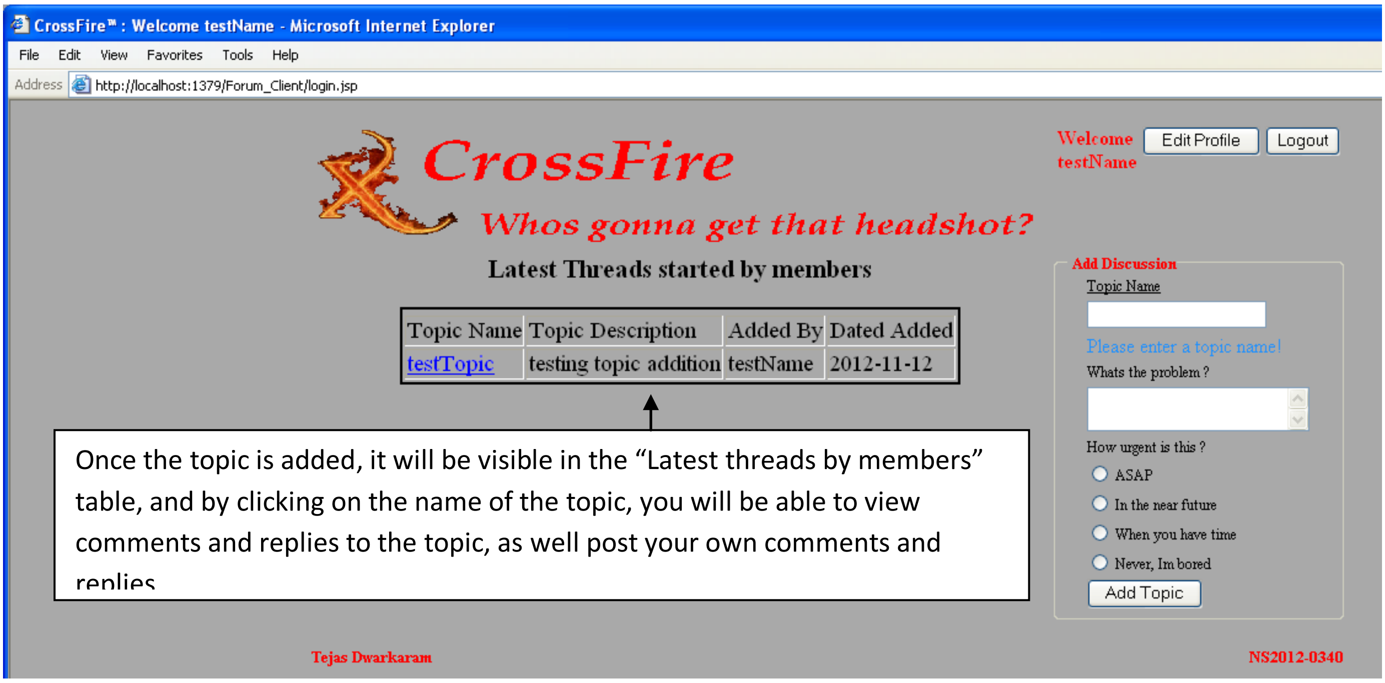
Topic Name: testTopic

Whats the problem ? testing topic addition

How urgent is this ?

○ ASAP
○ In the near future
● When you have time
○ Never, Im bored

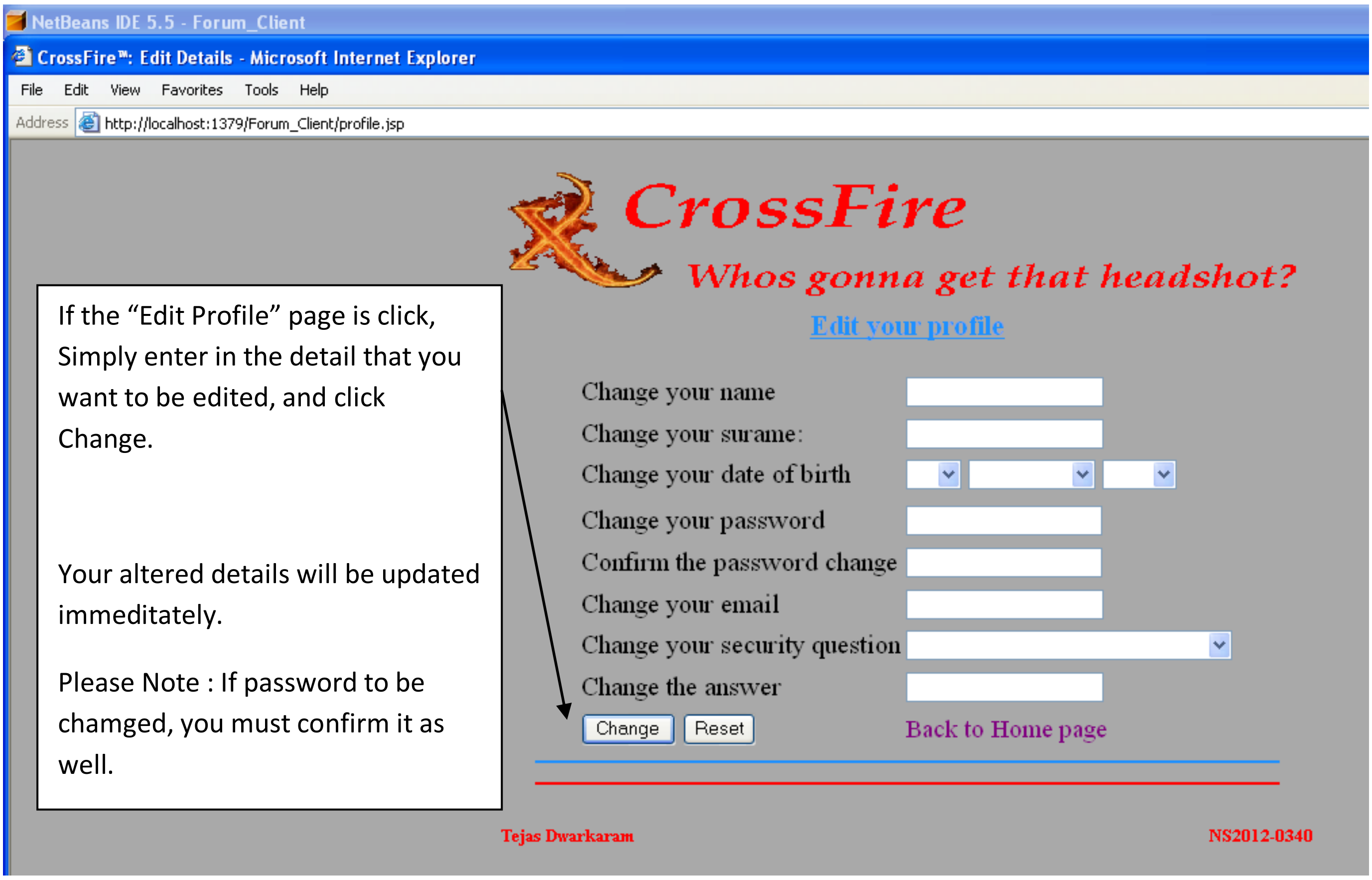
When adding a topic, enter the name of the topic, the detailed description of the topic as well as the urgency of the topic and click “Add Topic”



Latest Threads started by members

Topic Name	Topic Description	Added By	Dated Added
testTopic	testing topic addition	testName	2012-11-12

Once the topic is added, it will be visible in the “Latest threads by members” table, and by clicking on the name of the topic, you will be able to view comments and replies to the topic, as well post your own comments and replies



NetBeans IDE 5.5 - Forum_Client
CrossFire™: Edit Details - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:1379/Forum_Client/profile.jsp
CrossFire
Whos gonna get that headshot?
Change
Reset
Back to Home page
Tejas Dwarkaram
NS2012-0340

Edit your profile

Change your name	
Change your surame:	
Change your date of birth	
Change your password	
Confirm the password change	
Change your email	
Change your security question	
Change the answer	

If the “Edit Profile” page is click, Simply enter in the detail that you want to be edited, and click Change.

Your altered details will be updated immeditately.

Please Note : If password to be chamged, you must confirm it as well.

In the hope that my application helps you to solve all the problems you may come across, i ask that if any further assistance is required please feel free to contact me.

Source Code

Forum_Service

ForumWebService.java

```
/*
 * @author Tejas Dwarkaram
 * @version 1.5.0_9
 * @since 17 October 2012, 01:58
 *
 *
 *  fileName    ForumWebService.java
 *  description Creating the java file for the web service
 *              to hold all the methods the client can
 *              access, as well as access the database
 */


package ForumService;


import java.io.FileWriter;
import java.io.IOException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Properties;
import javax.jws.Oneway;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import javax.mail.MessagingException;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMessage.RecipientType;


/**
 *
 * @author Students
 */
@WebService()
public class ForumWebService {
   //creating a new instance of the ConnectionBEan class
   ConnectionBean connection = new ConnectionBean();
   /**
    * Web service operation
    */
   @WebMethod
   public boolean checkLogin(@WebParam(name = "username") String username,
@WebParam(name = "password") String password) {
      //creating a string variable to hold a default value
      String exists = "";
      //creating a new ResultSet
      ResultSet result;


      try {
        //Creating a new prepared statment to select the username and password and see if they
correspond
        PreparedStatement loginStatement = connection.getConnection().prepareStatement("SELECT
users_name, users_password FROM users WHERE users_name = ? AND users_password = ?");
        //setting the first placeholder to the username recieved from the client
        loginStatement.setString(1, username);
        //setting the first placeholder to the password recieved from the client
        loginStatement.setString(2, password);
        //executing the prepared statement and storing the results in the ResultSet
        result = loginStatement.executeQuery();


        //checking that the result set hold values
        while(result.next()){
          //getting the username value in the field returned
          exists = result.getString("users_name");
        }
        //closing the result set
        result.close();
        //closing the prepared statement
        loginStatement.close();
        //checking if the exists variable contains characters
        if(exists.length() < 1){
          //returning true if it did
          return false;
        }else{
          //returning false if it didnt
          return true;
        }
```

```
            //catching a SQL exception if it occurs
        } catch(SQLException sqle) {
            System.err.println("WebServiceError -> SQL Exception - on checking login : " + sqle.toString());
        }
        return false;
    }


    /**
     * Web service operation
     */
    @WebMethod
    public String getQuestion(@WebParam(name = "userEmail") String userEmail) {
        //creating a String variable to hold a default value
        String question = "NoQ";
        try {
            //Creating a new prepared statment to select the secQ field where the email corresponds to
the value recieved
            PreparedStatement queStatement = connection.getConnection().prepareStatement("SELECT
user_secQ FROM users WHERE users_email = ?");
            //setting the first placeholder to the userEmail variable recieved from the client
            queStatement.setString(1, userEmail);
            //storing the results of the prepared statement in a resultSet
            ResultSet rs = queStatement.executeQuery();


            //storing the question if the result set contains any values
            while(rs.next()){
                question = rs.getString("user_secQ");
            }
            //closing the result set
            rs.close();
            //closing the prepared statement
            queStatement.close();
            //using the trim() method to remove whitespaces
            question = question.trim();
            //retrieving and reseting the questions according to what they were stored as
            if(question.equals("mum")) {
                question = "What is your mums maiden name?";
            } else if(question.equals("pet")) {
                question = "What is your first pets name?";
            } else if(question.equals("teacher")) {
                question = "What is your first teachers name?";
            } else if(question.equals("town")) {
                question = "What is your home towns name?";
            } else if(question.equals("phone")) {
                question = "What is the name of your fav phone?";
            }
        //catching a SQL Exception if an error occurs
        } catch (SQLException ex) {
            System.err.println("WebServiceError -> SQL Exception - on getting question :" + ex.toString());
        }
        //returning the question variable
        return question;
    }


    /**
```

```
     * Web service operation
     */
    @WebMethod
    public boolean getAnswer(@WebParam(name = "userAns") String userAns) {
        //creating the default variables to hold the values that are going to be retrieved
        String password = "NoPassword";
        String email = "default@localhost";
        try {
            //Creating a new prepared statment to select all fields where the answer corresponds to the value recieved
            PreparedStatement ansStatement = connection.getConnection().prepareStatement("SELECT * FROM users WHERE user_secA = ?");
            //setting the first placeholder to the userAns variable recieved from the client
            ansStatement.setString(1, userAns);
            //storing the results of the prepared statement in a resultSet
            ResultSet rs = ansStatement.executeQuery();

            //storing the email and password if the result set contains any values
            while(rs.next()){
                password = rs.getString("users_password");
                email = rs.getString("users_email");
            }
            //closing the result set
            rs.close();
            //closing the prepared statement
            ansStatement.close();

            if(password.equals("")){
```

```
        //if they password contains nothing false is returned
        return false;
        //if it does the JavaMail sends an email to the email address retrieved
    }else{
        //creating a new date instance and formating it to time
        Properties props = System.getProperties();
        props.put("mail.stmp.host", "127.0.0.1");


        //craeting a new session
        Session session = Session.getDefaultInstance(props, null);
        //creating a new instance of the MimieMessage class
        MimeMessage message = new MimeMessage(session);


        //setting the values required and sending the email
        message.setSubject("Password Recovery - CrossFire Forums");
        //setting the contents of the email to include the forgotten password
        message.setText("Good day. This is a recovery email for your lost password.\nYour
password can be reset on next login.\nYour current password is: " + password + "\nSorry for any
inconvinience caused.\nRegards \nSystem Admin\nTejas Dwarkaran\nCrossFire Team");
        //setting the email address the user will see it was sent from
        message.setFrom(new InternetAddress("administrator@crossfire.net"));
        //setting the recipient address to the email address retrieved
        message.addRecipient(RecipientType.TO, new InternetAddress(email));


        //sending the message
        Transport.send(message);
        //returning true
        return true;
```

```
            }

            //catching a SQL or MEssaging exception that might occur

        } catch (SQLException ex) {

            System.err.println("WebServiceError -> SQL Exception - on sending email : " + ex.toString());

        }catch(MessagingException me) {

            System.err.println("WebServiceError -> Messaging Exception - on sending email : " +
me.toString());

        }

        //returning false by default

        return false;

    }


    /**

     * Web service operation

     */

    @WebMethod

    public boolean checkEmail(@WebParam(name = "userEmail") String userEmail,
@WebParam(name = "userName") String userName) {

        try {

            //creating a prepared statement to select all fields where the user name and email address
correspond

            PreparedStatement emailStatement =
connection.getConnection().prepareStatement("SELECT * FROM users WHERE users_name = ? AND
users_email = ?");

            //setting the first placeholder to the userName variable recieved from the client

            emailStatement.setString(1, userName);

            //setting the second placeholder to the userEmail variable recieved from the client

            emailStatement.setString(2, userEmail);

            //storing the values if the prepared statment return values
            ResultSet rs = emailStatement.executeQuery();


            //if the result set contains values returns true else false
            if(rs.next()){
                //closing the result set
                rs.close();
                //closing the prepared statement
                emailStatement.close();
                return true;
            }else {
                rs.close();
                emailStatement.close();
                return false;
            }
            //catching a SQL Exception that might occur
        } catch (SQLException ex) {
            System.err.println("WebServiceError -> SQL Exception - on checking email : " + ex.toString());
        }
        //returning false by default
        return false;
    }


    /**
     * Web service operation
     */
    @WebMethod
    public boolean addUser(@WebParam(name = "name") String name, @WebParam(name =
"surname") String surname, @WebParam(name = "dob") String dob,@WebParam(name =
"password") String password, @WebParam(name = "email") String email, @WebParam(name = 
"secQ") String secQ, @WebParam(name = "secA") String secA) {

        try {

            int addRows;

            String emailExist = "";

            //creating a prepared statement to select email address that may be the same as the variable 
recieved from the client

            PreparedStatement prepState = connection.getConnection().prepareStatement("SELECT 
users_email FROM users WHERE users_email = ?");

            //setting the first placeholder to the email variable recieved from the client

            prepState.setString(1, email);

            //storing the returned results from the statement in a resultset

            ResultSet rs = prepState.executeQuery();


            //checking if there are any values in the result set

            //to check for matches

            while(rs.next()){

                emailExist = rs.getString("users_email");

            }


            //closing the resultset

            rs.close();

            //closing the prepared statement

            prepState.close();

            /*

             *checking if the string variable emailExists contains anything

             */

            if(emailExist.equals("")){

                //creating another prepared statement to add the user details to the database
                prepState = connection.getConnection().prepareStatement("INSERT INTO
users(users_name, users_surname, users_dateOfBirth, users_password, users_email, user_secQ,
user_secA) VALUES( ?,?,?,?,?,?,? )");

                prepState.setString(1, name);

                prepState.setString(2, surname);

                prepState.setString(3, dob);

                prepState.setString(4, password);

                prepState.setString(5, email);

                prepState.setString(6, secQ);

                prepState.setString(7, secA);

                //executing the preparedStatement

                addRows = prepState.executeUpdate();

                //returning true

                return true;

            }else{

                return false;

            }

            //catching any SQL Execptions that might occur

        } catch(SQLException sql) {

            System.out.println("WebServiceError -> SQL Exception - on adding user : " + sql.toString());

        }

        return false;

     }




     /**

      * Web service operation

      */
    @WebMethod

    public boolean addTopic(@WebParam(name = "user") String user, @WebParam(name = 
"topicName") String topicName, @WebParam(name = "topicDesc") String topicDesc, 
@WebParam(name = "topicDate") String topicDate) {

        try {

            int addRows;

            //creating a prepared statement to insert topic details in to the database

            PreparedStatement prepState = connection.getConnection().prepareStatement("INSERT INTO 
topics(topic_name, topic_decs, topic_user, topic_date) VALUES( ?,?,?,? )");

            prepState.setString(1, topicName);

            prepState.setString(2, topicDesc);

            prepState.setString(3, user);

            prepState.setString(4, topicDate);

            //executing the insert

            addRows = prepState.executeUpdate();

            //closing the prepared statment

            prepState.close();

            //returning true

            return true;

            //catching any SQL Exceptions that might occur

        } catch(SQLException sql) {

            System.out.println("WebServiceError -> SQL Exception - adding topic : " + sql.toString());

        }

        return false;

    }


    /**

     * Web service operation

     */
    @WebMethod

    public String[] getTopics() {

        //creating a new ArrayList to store the values retrieved from the database

        ArrayList<String> listTopics = new ArrayList<String>();



        try {

            //creating the prepared statement to get all the topics

            PreparedStatement prepState = connection.getConnection().prepareStatement("SELECT *
FROM topics");

            //executing the prepared statment and storing the results in a result set

            ResultSet result = prepState.executeQuery();



            /*

             *adding each row of data to the arraylist

             */

            while(result.next()){

                listTopics.add(result.getString("topic_id") + "," + result.getString("topic_name") + "," +
result.getString("topic_decs") + "," + result.getString("topic_user") +  "," +
result.getString("topic_date"));

            }

            //closing the result set

            result.close();

            //closing the prepared statement

            prepState.close();

            //catching any SQL Exceptions that might occur

        } catch(SQLException sqle) {

            System.err.println("WebServiceError -> SQL Exception - getting topics : " + sqle.toString());

        }

        //sending the arraylist to the listToString method and returning that
        return listToString(listTopics);
    }

    /*
     * @name listToString
     * @param List<String>
     *
     * @description method to convert any List object
     *             to a String array
     */
    private String[] listToString(List<String> list) {
        //creating a new String Array with the size of the List recieved
        String[] stringArray = new String[list.size()];
        //storing each value of the list in the array
        for(int x = 0; x != list.size(); x++) {
            stringArray[x] = list.get(x);
        }
        //returning the string array
        return stringArray;
    }

    /**
     * Web service operation
     */
    @WebMethod
    public String getTopicDescription(@WebParam(name = "topicName") String topicName) {
        //creating a new Result Set
        ResultSet result;

        //creating a default string variable

        String topicDescription = "";


        try {

            //creating a new Prepared Statement to get the details of a specific topic

            PreparedStatement prepState = connection.getConnection().prepareStatement("SELECT *
      FROM topics WHERE topic_name = ?");

            //making the first placholders value of the parameter recieved

            prepState.setString(1, topicName);

            //executing the prepared statement and storing the values in a result set

            result = prepState.executeQuery();


            /*

             * While there are values in teh result set

             * Storing the value of the topic_decs column in the

             *      topicDescription variable

             */

            while(result.next()){

              topicDescription = result.getString("topic_decs");

            }


            //closing the result set

            result.close();

            //closing the prepared statement

            prepState.close();

            //catching any Sql exception that might occure

        } catch(SQLException sqle) {
            System.err.println("WebServiceError -> SQL Exception - on getting topic description : " +
sqle.toString());

        }

        //returning the topicDescription retrieved from the database

        return topicDescription;

    }


    /**

     * Web service operation

     */

    @WebMethod

    public String getTopicDatePosted(@WebParam(name = "topicName") String topicName) {

        //Creating a new result set

        ResultSet result;

        String topicDate = "";


        try {

            //creating a new Prepared Statement to get the details of a specific topic

            PreparedStatement prepState = connection.getConnection().prepareStatement("SELECT *
FROM topics WHERE topic_name = ?");

            //making the first placholders value of the parameter recieved

            prepState.setString(1, topicName);

            //executing the prepared statement and storing the values in a result set

            result = prepState.executeQuery();


            /*

             * While there are values in teh result set

             * Storing the value of the topic_date column in the
```

```
         *    topicDate variable
         */
        while(result.next()){
           topicDate = result.getString("topic_date");
        }

        //closing the result set
        result.close();
        //closing the prepared statement
        prepState.close();
        //catching any Sql exception that might occur
      } catch(SQLException sqle) {
        System.err.println("WebServiceError -> SQL Exception - on getting topic date : " +
sqle.toString());
      }
      return topicDate;
   }

   /**
    * Web service operation
    */
   @WebMethod
   public void editName(@WebParam(name = "newName") String newName, @WebParam(name =
"userName") String userName) {
      try {
        //creating the prepared statement to update the users table
        PreparedStatement prepState = connection.getConnection().prepareStatement("UPDATE
users SET users_name  = ? WHERE users_name = ?");
        //setting the first placeholder to the new value parameter recieved
```

```
            prepState.setString(1, newName);

            //setting the second parameter to the users name parameter recieved

            prepState.setString(2, userName);

            //executing the prepared statement

            prepState.executeUpdate();

            //closing the prepared statement

            prepState.close();

            //catching any SQL Exception that occurs

        } catch(SQLException sqle) {

            System.err.println("WebServiceError -> SQL Exception - on updating name : " +
sqle.toString());

        }

    }



    /**

     * Web service operation

     */

    @WebMethod

    public void editSurname(@WebParam(name = "newSurname") String newSurname,
@WebParam(name = "userName") String userName) {

        try {

            //creating the prepared statement to update the users table

            PreparedStatement prepState = connection.getConnection().prepareStatement("UPDATE
users SET users_surname  = ? WHERE users_name = ?");

            //setting the first placeholder to the new value parameter recieved

            prepState.setString(1, newSurname);

            //setting the second parameter to the users name parameter recieved

            prepState.setString(2, userName);

            //executing the prepared statement
            prepState.executeUpdate();

            //closing the prepared statement

            prepState.close();

            //catching any SQL Exception that occurs

        } catch(SQLException sqle) {

            System.err.println("WebServiceError -> SQL Exception - on updating surname : " +
sqle.toString());

        }

    }



    /**

     * Web service operation

     */

    @WebMethod

    public void editDob(@WebParam(name = "newDob") String newDob, @WebParam(name =
"userName") String userName) {

        try {

            //creating the prepared statement to update the users table

            PreparedStatement prepState = connection.getConnection().prepareStatement("UPDATE
users SET users_dateOfBirth  = ? WHERE users_name = ?");

            //setting the first placeholder to the new value parameter recieved

            prepState.setString(1, newDob);

            //setting the second parameter to the users name parameter recieved

            prepState.setString(2, userName);

            //executing the prepared statement

            prepState.executeUpdate();

            //closing the prepared statement

            prepState.close();

            //catching any SQL Exception that occurs
        } catch(SQLException sqle) {

            System.err.println("WebServiceError -> SQL Exception - on updating dob : " + sqle.toString());

        }

    }


    /**

     * Web service operation

     */

    @WebMethod

    public void editPassword(@WebParam(name = "newPass") String newPass, @WebParam(name =
"userName") String userName) {

        try {

            //creating the prepared statement to update the users table

            PreparedStatement prepState = connection.getConnection().prepareStatement("UPDATE
users SET users_password  = ? WHERE users_name = ?");

            //setting the first placeholder to the new value parameter recieved

            prepState.setString(1, newPass);

            //setting the second parameter to the users name parameter recieved

            prepState.setString(2, userName);

            //executing the prepared statement

            prepState.executeUpdate();

            //closing the prepared statement

            prepState.close();

            //catching any SQL Exception that occurs

        } catch(SQLException sqle) {

            System.err.println("WebServiceError -> SQL Exception - on updating pass : " + sqle.toString());

        }

    }
    /**
     * Web service operation
     */
    @WebMethod
    public void editSecQ(@WebParam(name = "newQues") String newQues, @WebParam(name =
"userName") String userName) {
        try {
            //creating the prepared statement to update the users table
            PreparedStatement prepState = connection.getConnection().prepareStatement("UPDATE
users SET user_secQ  = ? WHERE users_name = ?");
            //setting the first placeholder to the new value parameter recieved
            prepState.setString(1, newQues);
            //setting the second parameter to the users name parameter recieved
            prepState.setString(2, userName);
            //executing the prepared statement
            prepState.executeUpdate();
            //closing the prepared statement
            prepState.close();
            //catching any SQL Exception that occurs
        } catch(SQLException sqle) {
            System.err.println("WebServiceError -> SQL Exception - on updating secQ : " + sqle.toString());
        }
    }

    /**
     * Web service operation
     */
    @WebMethod
    public void editSecA(@WebParam(name = "newAns") String newAns, @WebParam(name = "userName") String userName) {
        try {
            //creating the prepared statement to update the users table
            PreparedStatement prepState = connection.getConnection().prepareStatement("UPDATE users SET user_secA  = ? WHERE users_name = ?");
            //setting the first placeholder to the new value parameter recieved
            prepState.setString(1, newAns);
            //setting the second parameter to the users name parameter recieved
            prepState.setString(2, userName);
            //executing the prepared statement
            prepState.executeUpdate();
            //closing the prepared statement
            prepState.close();
            //catching any SQL Exception that occurs
        } catch(SQLException sqle) {
            System.err.println("WebServiceError -> SQL Exception - on updating secA : " + sqle.toString());
        }
    }


    /**
     * Web service operation
     */
    @WebMethod
    public void editEmail(@WebParam(name = "newEmail") String newEmail, @WebParam(name = "userName") String userName) {
        try {
            //creating the prepared statement to update the users table
            PreparedStatement prepState = connection.getConnection().prepareStatement("UPDATE users SET users_email  = ? WHERE users_name = ?");

            //setting the first placeholder to the new value parameter recieved

            prepState.setString(1, newEmail);

            //setting the second parameter to the users name parameter recieved

            prepState.setString(2, userName);

            //executing the prepared statement

            prepState.executeUpdate();

            //closing the prepared statement

            prepState.close();

            //catching any SQL Exception that occurs

        } catch(SQLException sqle) {

            System.err.println("WebServiceError -> SQL Exception - on updating secA : " + sqle.toString());

        }

    }


    /**

     * Web service operation

     */

    @WebMethod

    public String[] getComments(@WebParam(name = "topicName") String topicName) {

        ArrayList<String> listComments = new ArrayList<String>();

        try {

            int topicId = 0;

            //creating a prepared statement to select the topic id from the topics table where the names correpond

            PreparedStatement prepState = connection.getConnection().prepareStatement("SELECT topic_id FROM topics WHERE topic_name = ?");

            //setting the place holder to the parameter recieved
            prepState.setString(1, topicName);
            // storing the results of the prepared statement in a result set
            ResultSet result = prepState.executeQuery();


            /*
             *While the result set contains values
             *Storing the topic ID in the topic ID variable
             */
            while(result.next()){
               topicId = result.getInt("topic_id");
            }
            //creating another prepared statement to select all the comments form the comments table where the topic id correspond
            prepState = connection.getConnection().prepareStatement("SELECT * FROM comment WHERE topic_id = ?");
            //setting the first placeholder to the topic ID that was retrieved above
            prepState.setInt(1, topicId);
            //storing the results of the prepared statement into the result set
            result = prepState.executeQuery();


            /*
             * While the result set contains values
             * Storing the values in list variable
             */
            while(result.next()){
               listComments.add(result.getString("comment_id") + "," + result.getString("users_name") + "," + result.getString("user_text") + "," + result.getString("comment_date"));
            }
```

```
            //closing the result set

            result.close();

            //closing the prepared statement

            prepState.close();


        } catch(SQLException sqle) {

            System.err.println("WebServiceError -> SQL Exception - on getting comments : " +
sqle.toString());

        }

        //sending the arraylist to the listToString method and returning that

        return listToString(listComments);

    }


    /**

     * Web service operation

     */

    @WebMethod

    public String[] getReplies(@WebParam(name = "commentId") String commentId) {

        //creating a new ArrayList to store the values retrieved from the database

        ArrayList<String> listComments = new ArrayList<String>();

        try {

            //creating the prepared statement to get all the replies where the comment id's correspond

            PreparedStatement prepState = connection.getConnection().prepareStatement("SELECT *
FROM replies WHERE comment_id = ?");

            // setting the first placeholder to the parameter recieved

            prepState.setInt(1, Integer.parseInt(commentId));

            //executing the prepared statment and storing the results in a result set

            ResultSet result = prepState.executeQuery();
            /*
             *adding each row of data to the arraylist
             */
            while(result.next()){
                listComments.add(result.getString("users_name") + "," + result.getString("user_reply") + ","
+ result.getString("reply_date"));
            }
            //closing the result set
            result.close();
            //closing the prepared statement
            prepState.close();
            //catching any SQL Exceptions that might occur
        } catch(SQLException sqle) {
            System.err.println("WebServiceError -> SQL Exception - on getting comments : " +
sqle.toString());
        }
        return listToString(listComments);
    }


    /**
     * Web service operation
     */
    @WebMethod
    @Oneway
    public void postComments(@WebParam(name = "userName") String userName,
@WebParam(name = "userComment") String userComment, @WebParam(name = "date") String
date, @WebParam(name = "topicId") int topicId) {
        SimpleDateFormat sdf = new SimpleDateFormat("hh:mm:SS");
        Date curDate = new Date();

        String time = sdf.format(curDate);

        try {

            //creating a prepared statment to insert a new record in the comments table

            PreparedStatement prepState = connection.getConnection().prepareStatement("INSERT INTO
comment(users_name, user_text, comment_date, topic_id) VALUES( ?,?,?,? )");

            //setting the name

            prepState.setString(1, userName);

            //setting the comment

            prepState.setString(2, userComment);

            //setting the date

            prepState.setString(3, date);

            //setting the topicId

            prepState.setInt(4, topicId);

            //executing the insert

            prepState.executeUpdate();

            //closing the prepared statement

            prepState.close();


            FileWriter fileOut = new FileWriter("C:/CrossFireLog.xflog", true);

            fileOut.write("-_-_-_-_-_-_-_-_-_\n");

            fileOut.write(date + " - " + time + " = User : " + userName + "\n");

            fileOut.write("\t Comment: " + userComment + " || On topic ID : " + topicId + "\n");

            fileOut.write("-_-_-_-_-_-_-_-_-_\n");


            fileOut.close();

        } catch(SQLException sqle) {
            System.err.println("WebServiceError -> SQL Exception - on posting comments : " +
sqle.toString());

        } catch(IOException ioe){

            System.err.println("WebServiceError -> IOException - on posting comments : " +
ioe.toString());

        }

    }


    /**

     * Web service operation

     */

    @WebMethod

    @Oneway

    public void postReplies(@WebParam(name = "userName") String userName, @WebParam(name
= "userReply") String userReply, @WebParam(name = "date") String date, @WebParam(name =
"commentId") int commentId) {

        SimpleDateFormat sdf = new SimpleDateFormat("hh:mm:SS");

        Date curDate = new Date();

        String time = sdf.format(curDate);


        try {

            //creating a prepared statment to insert a new record in the comments table

            PreparedStatement prepState = connection.getConnection().prepareStatement("INSERT INTO
replies(users_name, user_reply, reply_date, comment_id) VALUES( ?,?,?,? )");

            //setting the name

            prepState.setString(1, userName);

            //setting the replyText

            prepState.setString(2, userReply);

            //setting the date
```

```
      prepState.setString(3, date);

      //setting the commentId

      prepState.setInt(4, commentId);

      //executing the insert

      prepState.executeUpdate();

      //closing the prepared statement

      prepState.close();



      FileWriter fileOut = new FileWriter("C:/CrossFireLog.xflog", true);

      fileOut.write("-_-_-_-_-_-_-_-_-_\n");

      fileOut.write(date + " - " + time + " = User : " + userName + "\n");

      fileOut.write("\t Reply: " + userReply + " || On comment ID : " + commentId + "\n");

      fileOut.write("-_-_-_-_-_-_-_-_-_\n");



      fileOut.close();

    } catch(SQLException sqle) {

      System.err.println("WebServiceError -> SQL Exception - on posting replies : " +
sqle.toString());

    } catch(IOException ioe){

      System.err.println("WebServiceError -> IOException - on posting replies : " + ioe.toString());

    }

  }

}
```

ConnectionBean.java

```
package ForumService;

/*

 * @author Tejas Dwarkaram

 * @version 1.5.0_9

 * @since 21 October 2012, 10:51

 *

 *

 *  fileName    ConnectionBean.java

 *  description A java bean to initiate a

 *              connectiong to the database

 */



//importing the relevant libraries

import java.io.*;

import java.sql.*;


/*

 *          Creating the bean class and implementing the serializable

 *          class

 */

public class ConnectionBean implements Serializable {

   private Connection connection;

   private static String dbURL = "jdbc:odbc:forum_db";

   private static String driver = "sun.jdbc.odbc.JdbcOdbcDriver";
```

```
  /*
   *  Creating the constructor for the bean
   */
public ConnectionBean() {
  makeConnection();
}

  /*
   *  Creating the method to check if a connection is on
   *  @param null
   *  @return Connection type object
   */
public Connection getConnection() {
  //checking if the connection variable has been created or not
  if(connection == null) {
    makeConnection();
  }
  return connection;
}

  /*
   *  Creating the method to stop the connection
   *  @param Boolean variable of state of connection
   *  @return Null
   */
public void setCloseConnection(boolean close) {
  //if the connection is true
```

```
        if(close) {
            try {
                //close the connection
                connection.close();
            } catch(Exception e) {
                connection = null;
            }
        }
    }


        /*
         *   Creating the method to create a connection
         *   @param null
         *   @return null
         */
    public void makeConnection() {
        try {
            Class.forName(driver);
            connection = DriverManager.getConnection(dbURL);
        } catch(SQLException sqle) {
            System.out.println("Web Service Error -> SQL Exception on making db connection : " + sqle);
        } catch(ClassNotFoundException cnfe) {
            System.out.println("Web Service Error -> ClassNotFound Exception on making db connection : 
" + cnfe);
        }
    }
}
```

Source Code

Forum_Client

index.jsp

```
<!--
-- Author              Tejas Dwarkaram
-- FileName             index.jsp
-- Date               2012/11/01

-- File Description       JSP File that is the main page
                        /index page of the client website
                        contains little detail about the
                        forum, as well as facility to go sign
                        up, and login as well as to recover
                        passwords
-->

<%-- Creating the taglib tag to import the custom tag used for the banner --%>
<%@taglib uri="//WEB-INF/tlds/LogoFormatTag.tld" prefix="ban" %>
<html>
  <head>
    <title>CrossFire&trade;</title>
    <!-- setting a new session object that will be used in the recovery process -->
    <%session.setAttribute("step", "0");%>
  </head>

  <body style="background:darkgray">
    <table align="center" style="width:600px;">
      <tr>
        <!-- Using the custom tag that was imported in the taglib tag -->
        <td valign="top">
          <ban:BannerTag/>
        </td>
        <td valign="top">
          <form action="LoginServlet" method="post">
            <%
                        /*
                        *Creating the string variables to check if any errors
                        *have occured, and display them if so. By using
                        *parameters sent thru the header fields from the other
                        *pages and servlets
                         */
                        String error = request.getParameter("error") == null ? "" : "Error on username or
password, please try again";
                        String logged = request.getParameter("logged") == null ? "" : "You arent logged in!";
                        String logoff = request.getParameter("logoff") == null ? "" :
request.getParameter("logoff");
                        /*
                        *If statment to check if the user had clicked the
                        *logoff button on the login page, which then end the
                        *session by removing the attribute linked to the user
                        *that logged in.
                         */
                        if(logoff.equals("true")){
                          session.removeAttribute("user");
                        }%>

            <!-- Creating the simple form for users
```

```
                    to log into the website as well as see other

                    options if they arent users

                    or have forgotten their passwords-->

                <table style="color:red;font-weight:bold;border-style:solid;border-
width:thin;color:red">

                   <tr>

                      <td>

                         Username

                      </td>

                      <td>

                         <input type="text" size=18 maxlength=15 name="username"/>

                      </td>

                   </tr>

                   <tr>

                      <td>

                         Password

                      </td>

                      <td>

                         <input type="password" size=18 maxlength=15 name="password"/>

                      </td>

                   </tr>

                   <tr><td colspan="2"><%out.println(error + " " + logged);%></td></tr>

                   <tr>

                      <td>

                         <input type="submit" value="Login"/>

                      </td>

                      <td>
```

```
            Forgot password ? <a href="remember.jsp" style="text-decoration: none" title="Click here we'll help you remember !">Click to recover</a>
          </td>
        </tr>
        <tr>
          <td colspan="3">
            Not signed up ? <a href="register.jsp" style="text-decoration: none" title="Not a user ? Sign up today! Dont hesitate">Register Here</a>
          </td>
        </tr>
      </table>
    </form>
  </td>
</tr>
<tr>
  <td colspan="2" valign="top"></td>
</tr>
<!-- A small paragraph of text to tell the user
   what this website is all about-->
<tr>
  <td colspan="2" valign="top" align="center">
    <table style="border-style:solid;border-width:thin;color:red">
      <tr>
        <td>
          <h3 style="font-size:16pt;color:dodgerblue" >
            About CrossFire &trade;
          </h3>
        </td>
```

```
            </tr>
            <tr>
              <td>
                <p style="color:black; font-size:14pt">
                  So CrossFire ....

                  <b><u>What is CrossFire?</u></b>
                  <hr style="color:dodgerblue"/>
                  CrossFire is essentially a forum based web site, that allow users to <br/>
                  post about questions that they have which they, themselves, cannot answer.<br/>
                  So the topics will be put up for discussion, and other users, can log in<br/>
                  and assist you with these topics.<br/>

                  <hr style="color:dodgerblue"/>
                  The meaning behind "So who's gonna get that headshot?" is simple !<br/>
                  Basically, when trying to answer a topic, during the process of<br/>
                  replying and commenting, one can be seen as "firing" bullets at the <br/>
                  target(which is the topic at hand). And the person to actually solve the<br/>
                  discussion with have in turn "killed the topic", which in essence can be seen<br/>
                  as a Headshot!<br/><br/>



                  <hr style="color:dodgerblue"/>
                  So .... Do you have what it takes to get some Headshots !?
                </p>
                </td>
              </tr>
            </table>
          </td>
        </tr>
        <!-- A simple footer for the page -->
        <tr style="font-weight:bold;font-size:10pt;color:red;">
          <td valign="top">
            <p align="left">
              Tejas Dwarkaram
            </p>
          </td>
          <td valign="top">
            <p align="right">
              NS2012-0340
            </p>
          </td>
        </tr>
      </table>
    </body>
  </html>
```

register.jsp

```
<!--

-- Author              Tejas Dwarkaram

-- FileName             register.jsp

-- Date                2012/11/01


-- File Description       JSP File that will allow the user to

                       easily input their personal information

                       and complete the user registration process

-->


<%-- Creating the taglib tag to import the custom tag used for the banner --%>

<%@taglib uri="//WEB-INF/tlds/LogoFormatTag.tld" prefix="ban" %>

<html>


  <head>

    <title>Sign up for CrossFire&trade;</title>

  </head>


  <body style="background:darkgray">

    <%

    /*

    *Creating string variables to hold values if

    *any errors occur, which will be seen in the

    *header field

    */

    String invalid = "";

    String nameError = "";
    String surnameError = "";

    String monthError = "";

    String febError = "";

    String lengthError = "";

    String matchError = "";

    String emailError = "";

    String answerError = "";

    //checking if the header field for parameters that will show success, errors or email exists

    String errors = request.getParameter("errors") == null ? "" : request.getParameter("errors");

    String exists = request.getParameter("exist") == null ? "" : "The email address has been
registered before!";


    /*

    *checking the errors variable, which checks if the header for

    *that parameter contains numbers in correspondence to the error

    *occured in which field

     */

    if(errors.contains("0")){

      invalid = "Invalid charaters have been used";

    }

    if(errors.contains("1")){

      nameError = "Error on your name";

    }

    if(errors.contains("2")){

      surnameError = "Error on your surname";

    }

    if(errors.contains("3")){

      monthError = "This month has only 30 days";
    }
    if(errors.contains("4")){
      febError = "February has 28 days";
    }
    if(errors.contains("5")){
      lengthError = "Password is to short";
    }
    if(errors.contains("6")){
      matchError = "Passwords dont match";
    }
    if(errors.contains("7")){
      emailError = "Error in your email";
    }
    if(errors.contains("8")){
      answerError = "Answer cant contain errors";
    }
    %>
    <table align="center" style="width:600px;">
      <!-- Using the custom tag that was imported in the taglib tag -->
      <tr>
          <td valign="top" colspan="2">
              <ban:BannerTag/>
          </td>
      </tr>
      <tr>
          <td colspan="2" valign="top" align="center">
              <h1 style="color:dodgerblue">
```

```
  <u>Registration CrossFire&trade;</u>
</h1>
 <!-- Printing out an error if the email address is in use already -->
<h2 style="color:red"><%out.println(exists);%></h2>
<form action="RegisterServlet" method="post" style="font-size:15">
  <b>
    <table style="font-size:15pt">
      <tr>
        <td></td>
        <!-- Printing out an error if there are unknown characters present-->
        <td style="font-size:12pt; color:dodgerblue;"><%out.println(invalid);%></td>
      </tr>
      <tr>
        <td>Name:</td>
        <td><input type="text" name="name"/></td>
      </tr>
      <tr>
        <td></td>
        <!-- Printing out an a message if  there is an error -->
        <td style="font-size:12pt; color:dodgerblue;"><%out.println(nameError);%>
        </td>
      </tr>
      <tr>
        <td>Surame:</td>
        <td><input type="text" name="surname"/></td>
      </tr>
      <tr>
  <td></td>
  <!-- Printing out an a message if  there is an error -->
  <td style="font-size:12pt; color:dodgerblue;"><%out.println(surnameError);%>
  </td>
</tr>
<tr>
  <td>Date of birth:</td>
  <td>
    <select name="day">
      <%
      for(int x=1 ;  x<32 ; x++){
    out.print("<option value='" + x + "'>" + x + "</option>");
      }
      %>
    </select>
    <select name="month">
      <option value="01">January</option>
      <option value="02">Febuary</option>
      <option value="03">March</option>
      <option value="04">April</option>
      <option value="05">May</option>
      <option value="06">June</option>
      <option value="07">July</option>
      <option value="08">August</option>
      <option value="09">September</option>
      <option value="10">October</option>
      <option value="11">November</option>
```

```
                    <option value="12">December</option>
                 </select>
                 <select name="year">
                    <%
                    for(int y = 1900 ; y <2012 ; y++){
                 out.print("<option value='" + y + "'>" + y + "</option>");
                    }
                    %>
                 </select>
              </td>
           </tr>
           <tr>
              <td></td>
              <!-- Printing out an a message if  there is an error -->
              <td style="font-size:12pt;
color:dodgerblue;"><%out.println(monthError);%></td>
           </tr>
           <tr>
              <td></td>
              <!-- Printing out an a message if  there is an error -->
              <td style="font-size:12pt; color:dodgerblue;"><%out.println(febError);%></td>
           </tr>
           <tr>
              <td>Password:</td>
              <td><input type="password" name="pwd"/></td>
           </tr>
           <tr>
              <td></td>
```

```
         <!-- Printing out an a message if  there is an error -->
         <td style="font-size:12pt;
color:dodgerblue;"><%out.println(lengthError);%></td>
        </tr>
        <tr>
         <td>Confirm Password:</td>
         <td><input type="password" name="cnpwd"/></td>
        </tr>
        <tr>
         <td></td>
         <!-- Printing out an a message if  there is an error -->
         <td style="font-size:12pt;
color:dodgerblue;"><%out.println(matchError);%></td>
        </tr>
        <tr>
         <td>Email Address:</td>
         <td><input type="text" name="email"/></td>
        </tr>
        <tr>
         <td></td>
         <!-- Printing out an a message if  there is an error -->
         <td style="font-size:12pt;
color:dodgerblue;"><%out.println(emailError);%></td>
        </tr>
        <tr>
         <td>Security Question</td>
         <td>
          <select name="secq">
           <option value="pet">What is your first pets name?</option>
```

```
                      <option value="mum">What is your mums name?</option>

                      <option value="teacher">What is your first teachers name?</option>

                      <option value="town">What is your home towns name?</option>

                      <option value="phone">What is the name of your fav phone?</option>

                    </select>

                  </td>

                </tr>

                <tr>

                  <td>What is the answer to this question ?</td>

                  <td><input type="text" name="seca"/></td>

                </tr>

                <tr>

                  <td></td>

                  <!-- Printing out an a message if there is an error -->

                  <td style="font-size:12pt;
color:dodgerblue;"><%out.println(answerError);%></td>

                </tr>

                <tr>

                  <td style="font-size:14pt">

                    <input type="submit" value="Sign Up"/>

                    <input type="reset" value="Reset"/>

                  </td>

                  <td colspan="3">

                    <a href="index.jsp" style="text-decoration:none; font-size:14pt;">Back to
index page</a>

                  </td>

                </tr>

              </table>
```

```
                <hr style="color:dodgerblue" width="550px"/>
                <hr style="color:red" width="550px"/>
              </b>
            </form>
          </td>
        </tr>
        <!-- A simple footer -->
        <tr style="font-weight:bold;font-size:10pt;color:red;">
          <td valign="top">
            <p align="left">
              Tejas Dwarkaram
            </p>
          </td>
          <td valign="top">
            <p align="right">
              NS2012-0340
            </p>
          </td>
        </tr>
      </table>
    </body>
  </html>
```

newUser.jsp

```
<!--
-- Author            Tejas Dwarkaram
-- FileName            newUser.jsp
-- Date              2012/11/01

-- File Description       JSP File that will welcome a user
                    if all registration details were
                    correct
-->

<%-- Creating the taglib tag to import the custom tag used for the banner --%>
<%@taglib uri="//WEB-INF/tlds/LogoFormatTag.tld" prefix="ban" %>
<html>

  <!-- Setting the title of the page -->
  <head>
    <title>Welcome to CrossFire&trade;</title>
  </head>

  <body style="background:darkgray">
    <table align="center" style="width:600px;">
      <tr>
        <!-- Using the custom tag that was imported in the taglib tag -->
        <td valign="top" colspan="2">
          <ban:BannerTag/>
        </td>
```

```
</tr>
<tr>
  <!-- Writing out a welcome message to the user -->
  <td valign="top" align="center" colspan="2">
    <table style="color:red;font-weight:bold;border-style:solid;border-width:thin;color:red">
      <tr>
        <td>
          Congratulations !
          Welcome to CrossFire!
          We hope we help to solve all your problems, and good luck with you getting
          <u><big>Headshots!</big></u>
          You will now be redirected to the Home page
          Where you will conduct your login
          <hr/>
        </td>
      </tr>
      <tr>
        <td>
          <a href='index.jsp' style='text-decoration: none'>Click here to proceed!</a>
        </td>
      </tr>
    </table>
  </td>
</tr>
<!-- A simple footer for the page -->
<tr style="font-weight:bold;font-size:10pt;color:red;">
  <td valign="top">
```

```
                <p align="left">
                    Tejas Dwarkaram
                </p>
            </td>
            <td valign="top">
                <p align="right">
                    NS2012-0340
                </p>
            </td>
        </tr>
    </table>
  </body>
</html>
```

login.jsp

```
<!--
-- Author               Tejas Dwarkaram
-- FileName              login.jsp
-- Date                 2012/11/01

-- File Description       JSP File that is the home page
                        when a user logins in succesfully
                        and shows current topics, and allows
                        for new topics to be added.
-->

<%-- Creating the taglib tag to import the custom tag used for the banner --%>
<%@taglib uri="//WEB-INF/tlds/LogoFormatTag.tld" prefix="ban" %>
<html>
  <head>
    <title>
      <%
      /*
      *creating a string variable to retrieve the active session
      *object that contains a attribute, that signifies if the user
      *logged in or not
       */
      String username = (String)session.getAttribute("user");
      boolean allCorrect = true;
      out.println("CrossFire&trade; : Welcome " + username );
      %>
    </title>
</head>

<body style="background:darkgray">
    <%
    /*
    *checking if the user is logged in, else
    *redirecting to the index page,
    *and stating he needs to login
    */
    if(username == null || username == ""){
        response.sendRedirect("index.jsp?logged=false");
    }
    %>
    <table align="center" style="width:600px;">
        <tr>
            <!-- Using the custom tag that was imported in the taglib tag -->
            <td valign="top">
                <ban:BannerTag/>
            </td>
            <td valign="top">
                <table style="color:red;font-size:12pt;font-weight:bold;">
                    <tr>
                        <td valign="top">
                            Welcome <%out.println(username);%>
                        </td>
                        <td>
                            <form action="profile.jsp">
                                <input type="submit" value="Edit Profile" name=<%out.println(username);%>/>
                            </form>
                        </td>
                        <td>
                            <form action="index.jsp?logoff=true">
                                <input type="submit" value="Logout"/>
                            </form>
                        </td>

                    </tr>
                </table>
            </td>
        </tr>
        <tr align="center">
            <td valign="top">
                <h3 style="font-size:16pt" >
                    Latest Threads started by members
                </h3>
                <table border="2" style="font-size:14pt; border-color:black;">
                    <tr style="color:dodgerblue" >
                        <tr>
                            <td>Topic Name</td>
                            <td>Topic Description</td>
                            <td>Added By</td>
                            <td>Dated Added</td>
                        </tr>
```

```
<%
/*
*calling the webservice, and invoking the getTopics method
*which will display current topics if any exist
*/
try {
    //creating a new instance of the implementation of the web service
    ForumService.ForumWebServiceService forumWebServiceService = new ForumService.ForumWebServiceService_Impl();
    //using the implementation of the web service to get the active port used by the web service
    ForumService.ForumWebService forumWebServicePort = forumWebServiceService.getForumWebServicePort();
    //creating a string array to hold the values returned
    String topics[] = forumWebServicePort.getTopics();
    //for loop to get each element of the array returned
    for(int y = 0; y < topics.length; y++){
        //Creating an array of the individual values in the elements
        String[] allTopics = topics[y].split(",");
        out.println("<tr>");
        String topicId = "";
        int count = 0;
        //using a for loop to print out the contents in html table form
        for(int x = 0; x < allTopics.length; x++){
            count++;
            if(count == 1){
                topicId = allTopics[x];
            }
            if(x==1){
```

```
                                        out.println("<td><a href='topicspage.jsp?topic=" + allTopics[x] + "&topicId="
+ topicId + "'>" + allTopics[x] + "</td>");

                                    }else if(x>1){

                                        out.println("<td>" + allTopics[x] + "</td>");

                                    }

                                }

                                out.println("</tr>");

                            }

                            //catching any exception that might occure through the web service

                        } catch(javax.xml.rpc.ServiceException ex) {

                            System.err.println("ServiceException found on getting topic : " + ex.toString());

                        } catch(java.rmi.RemoteException ex) {

                            System.err.println("RemoteException found on getting topic : " + ex.toString());

                        } catch(Exception ex) {

                            System.err.println("UnknownException found on getting topic : " + ex.toString());

                        }

                        %>

                    </tr>

                </table>

            </td>

            <!-- Creating a new HTML table to house components

                that allow user to add a new topic for discussion-->

            <td valign="top" style="font-weight:bold;font-size:10pt">

                <form method="post">

                    <fieldset>

                        <legend style="color:red;">

                            Add Discussion

                        </legend>
```

```
              <table>
                <tr>
                  <td>
                    <u>Topic Name</u>
                  </td>
                </tr>
                <tr>
                  <td>
                    <input type="text" name="topicName"/>
                  </td>
                </tr>
                <%
                /*
                * Checking that the name field contains values
                */
                String topName = request.getParameter("topicName")== null ? "" :
request.getParameter("topicName");
                if(topName.length() < 1){
                  out.println("<tr>");
                  out.println("<td colspan='2' style='font-size:12pt; color:dodgerblue;'>Please
enter a topic name!</td>");
                  out.println("</tr>");
                  allCorrect = false;
                }
                %>
                <tr>
                  <td>
                    Whats the problem ?
```

```
  </td>
</tr>
<tr>
  <td>
    <textarea name="topicDesc"></textarea>
  </td>
</tr>
<%
/*
* Checking that the description field contains values
*/
String topDesc = request.getParameter("topicDesc")== null ? "  " : request.getParameter("topicDesc");
if(topDesc.length() < 1){
  out.println("<tr>");
  out.println("<td colspan='2' style='font-size:12pt; color:dodgerblue;'>Please enter a topic description!</td>");
  out.println("</tr>");
  allCorrect = false;
}
%>
<tr>
  <td>
    How urgent is this ?
  </td>
</tr>
<tr>
  <td>
```

```
      <input type="radio" name="urgency" value="Asap" > ASAP </input>
   </td>
</tr>
<tr>
   <td>
      <input type="radio" name="urgency" value="Future" > In the near future </input>
   </td>
</tr>
<tr>
   <td>
      <input type="radio" name="urgency" value="Time" > When you have time </input>
   </td>
</tr>
<tr>
   <td>
      <input type="radio" name="urgency" value="Bored" > Never, Im bored </input>
   </td>
</tr>
<%
/*
* Checking that the urgency field has been selected
*/
String topicUrgency = request.getParameter("urgency") == null ? "  " : request.getParameter("urgency");
if(topicUrgency.length() < 1){
   out.println("<tr>");
```

```
                  out.println("<td colspan='2' style='font-size:12pt; color:dodgerblue;'>Please
enter the urgency!</td>");
                  out.println("</tr>");
                  allCorrect = false;
                }
                %>
                <tr>
                  <td>
                    <input type="submit" value="Add Topic"/>
                  </td>
                </tr>
              </table>
            </fieldset>
          </form>
          <%
          /*
          *If all of the fields are correct
          *the current date is determined, and
          *the topic details are sent to the
          *web service, which adds the topic to
          *the table in the database
          */
          if(allCorrect){
            //creating a new instance of the SimpleDateFormat class to format the date
            java.text.SimpleDateFormat sdf = new java.text.SimpleDateFormat("yyyy-MM-dd");
            //getting the current sytem time and date
            java.util.Date date = new java.util.Date();
            //formatting the current date using the format method defined above
            String fullDate = sdf.format(date);

            try {

              //creating a new instance of the implementation of the web service

              ForumService.ForumWebServiceService forumWebServiceService = new
ForumService.ForumWebServiceService_Impl();

              //using the implementation of the web service to get the active port used by the web
service

              ForumService.ForumWebService forumWebServicePort =
forumWebServiceService.getForumWebServicePort();

              //creating a boolean variable to hold the returned "true" or "false "value"

              boolean done = forumWebServicePort.addTopic(username.trim(), topName.trim(),
topDesc.trim(), fullDate.trim());

              response.sendRedirect("login.jsp");

              //catching any exceptions that might be thrown

            } catch(javax.xml.rpc.ServiceException ex) {

              System.err.println("ServiceException found on adding topic : " + ex.toString());

            } catch(java.rmi.RemoteException ex) {

              System.err.println("RemoteException found on adding topic : " + ex.toString());

            } catch(Exception ex) {

              System.err.println("UnknownException found on adding topic : " + ex.toString());

            }

          }

          %>

        </td>

      </tr>

      <tr style="font-weight:bold;font-size:10pt;color:red;">

        <td valign="top">

          <p align="left">

            Tejas Dwarkaram
```

```
                </p>
              </td>
              <td valign="top">
                <p align="right">
                  NS2012-0340
                </p>
              </td>
            </tr>
          </table>
        </body>

      </html>
```

remember.jsp

```
<!--

-- Author              Tejas Dwarkaram

-- FileName             remember.jsp

-- Date                 2012/11/01


-- File Description       JSP File that will use the "step" session

                          and be used to help the user recover his

                          password by sending him/her their password

                          via an email

-->


<%-- Creating the taglib tag to import the custom tag used for the banner --%>

<%@taglib uri="//WEB-INF/tlds/LogoFormatTag.tld" prefix="ban" %>

<html>


  <head>

    <title>CrossFire&trade; Password Recovery</title>

  </head>


  <body style="background:darkgray">

    <table align="center" style="width:600px;">

      <%

      //Creating the String variable to store which step the user is currently on

      String step = request.getParameter("step") == null ? "0" : request.getParameter("step");

      %>

      <tr>

        <!-- Using the custom tag that was imported in the taglib tag -->
        <td valign="top" colspan="2">
          <ban:BannerTag/>
        </td>
      </tr>
      <tr>
        <td colspan="2" valign="top" align="center">
          <h3 style="font-size:15pt">
            <u>Password Recovery assistant</u>
          </h3>
          <b>
            <% //checking the contents of the step variable
            if(step.equals("0") || step.equals("1")){
              session.setAttribute("step", "1");
            %>
            <form action="RecoveryServlet" method="post">
              <fieldset style="align:top;width: 200px;" >
                <legend style="align:left;color: black; font-size:14pt">Verify Email
Address</legend>
                <table style="color:red;">
                  <tr>
                    <td>
                      <%
                      if(step.equals("1")){
                      %>
                      Ooops ! Please try that 1 more time..
                      <%
                      }
                      %>
```

```
  </td>
</tr>
<tr>
  <td style="font-size:12pt">
    Your user name
  </td>
</tr>
<tr>
  <td>
    <input type="text" size=18 maxlength=24 name="name"/>
  </td>
</tr>
<tr>
  <td style="font-size:12pt">
    Your email address
  </td>
</tr>
<tr>
  <td>
    <input type="text" size=18 maxlength=24 name="email"/>
  </td>
</tr>
<tr>
  <td>
    <input type="submit" value="Verify"/>
  </td>
</tr>
```

```
        </table>
    </fieldset>
</form>
<%
}else if(step.equals("2")){
session.setAttribute("step", "2");
%>
<form action="RecoveryServlet" method="post">
    <fieldset style="align:top;width: 200px;" >
        <legend style="align:left;color: black; font-size:14pt">Security Question</legend>
        <table style="color:red;">
            <tr>
                <td style="font-size:12pt">
                    <b><u>Your Security Question is</u></b>
                </td>
            </tr>
            <tr>
                <td>
                    <%
                    out.println(request.getParameter("que"));
                    %>
                </td>
            </tr>
            <tr>
                <td style="font-size:12pt">
                    Please enter your answer
                </td>
            </tr>
            <tr>
               <td>
                  <input type="text" size=18 maxlength=24 name="answer"/>
               </td>
            </tr>
            <tr>
               <td>
                  <input type="submit" value="Send"/>
               </td>
            </tr>
         </table>
   </fieldset>
</form>
<%
/*
 *checking if the step is on 3 which means the email
 *was sent successfully
 */
}else if(step.equals("3")){
%>
<form  method="post">
   <fieldset style="align:top;width: 200px;" >
      <legend style="align:left;color: black; font-size:14pt">Success!!</legend>
      <table style="color:red;">
         <tr>
            <td style="font-size:12pt">
```

```
                Your password has been successfully recovered !<br/>
                An email has been sent to your email address <br/>
                Your password can be changed on next login to <br/>
                further forgetfulness<br/>
                <a href="index.jsp"> Back to home page for login </a>
              </td>
            </tr>
          </table>
        </fieldset>
      </form>
      <%}%>
    </b>
  </td>
</tr>
<!-- A simple footer for the page -->
<tr style="font-weight:bold;font-size:10pt;color:red;">
  <td valign="top">
    <p align="left">
      Tejas Dwarkaram
    </p>
  </td>
  <td valign="top">
    <p align="right">
      NS2012-0340
    </p>
  </td>
</tr>
```

```
        </table>
    </body>
</html>
```

profile.jsp

```
<!--
-- Author              Tejas Dwarkaram
-- FileName              profile.jsp
-- Date                 2012/11/01

-- File Description       JSP File that will allow the user to
                       easily change their personal information
                       that was required on user registration
-->

<%-- Creating the taglib tag to import the custom tag used for the banner --%>
<%@taglib uri="//WEB-INF/tlds/LogoFormatTag.tld" prefix="ban" %>
<html>

  <head>
    <title>CrossFire&trade;: Edit Details</title>
  </head>

  <body style="background:darkgray">
    <%
    /*
    *Creating string variables to hold values if
    *any errors occur, which will be seen in the
    *header field
    */
    String invalid = "";
```

```
        String nameError = "";

        String surnameError = "";

        String monthError = "";

        String febError = "";

        String lengthError = "";

        String matchError = "";

        String emailError = "";

        String answerError = "";


        /*

        *Creating string variables to hold values if

        *the update the user requested was successful

        */

        String nameUpdate = "";

        String surnameUpdate = "";

        String dobUpdate = "";

        String passwordUpdate = "";

        String secQUpdate = "";

        String secAUpdate = "";

        String emailUpdate = "";

        //checking if the header field for parameters that will show success, errors or email exists

        String errors = request.getParameter("errors") == null ? "" : request.getParameter("errors");

        String updated = request.getParameter("updated") == null ? "" :
    request.getParameter("updated");

        String exists = request.getParameter("exist") == null ? "" : "The email address has been
    registered before!";


        /*
```

```
*checking the updated variable, which checks if the header for
*that parameter contains numbers in correspondence to the successful
*field updated
 */
if(updated.contains("1")){
   nameUpdate = "Name updated Successfully";
}
if(updated.contains("2")){
   surnameUpdate= "Surname updated Successfully";
}
if(updated.contains("3")){
   dobUpdate = "Date of birth updated Successfully";
}
if(updated.contains("4")){
   passwordUpdate = "Password updated Successfully";
}
if(updated.contains("5")){
   secQUpdate= "Security Question updated Successfully";
}
if(updated.contains("6")){
   secAUpdate = "Security Answer updated Successfully";
}
if(updated.contains("7")){
   emailUpdate = "Email updated Successfully";
}
/*
*checking the errors variable, which checks if the header for
```

```
*that parameter contains numbers in correspondence to the error
*occured in which field
 */
if(errors.contains("0")){
   invalid = "Invalid charaters have been used";
}
if(errors.contains("1")){
   nameError = "Error on your name";
}
if(errors.contains("2")){
   surnameError = "Error on your surname";
}
if(errors.contains("3")){
   monthError = "This month has only 30 days";
}
if(errors.contains("4")){
   febError = "February has 28 days";
}
if(errors.contains("5")){
   lengthError = "Password is to short";
}
if(errors.contains("6")){
   matchError = "Passwords dont match";
}
if(errors.contains("7")){
   emailError = "Error in your email";
}
```

```
if(errors.contains("8")){
   answerError = "Answer cant contain errors";
}
%>
<table align="center" style="width:600px;">
   <!-- Using the custom tag that was imported in the taglib tag -->
   <tr>
      <td valign="top" colspan="2">
         <ban:BannerTag/>
      </td>
   </tr>
   <tr>
      <td colspan="2" valign="top" align="center">
         <h1 style="color:dodgerblue">
            <u>Edit your profile</u>
         </h1>
         <!-- Printing out an error if the email address is in use already -->
         <h2 style="color:red"><%out.println(exists);%></h2>
         <form action="ProfileServlet" method="post" style="font-size:15">
            <b>
               <table style="font-size:15pt">
                  <tr>
                     <td></td>
                     <!-- Printing out an error if there are unknown characters present-->
                     <td style="font-size:12pt; color:dodgerblue;"><%out.println(invalid);%></td>
                  </tr>
                  <tr>
```

```
          <td>Change your name</td>
          <td><input type="text" name="name"/></td>
        </tr>
        <tr>
          <td></td>
          <!-- Printing out an a message if the update was successful or there is an error --
>
          <td style="font-size:12pt; color:dodgerblue;"><%out.println(nameError + "" +
nameUpdate);%></td>
        </tr>
        <tr>
          <td>Change your surame:</td>
          <td><input type="text" name="surname"/></td>
        </tr>
        <tr>
          <td></td>
          <!-- Printing out an a message if the update was successful or there is an error --
>
          <td style="font-size:12pt; color:dodgerblue;"><%out.println(surnameError + "" +
surnameUpdate);%></td>
        </tr>
        <tr>
          <td>Change your date of birth</td>
          <td>
            <select name="day">
              <option></option>
              <%
              for(int x=1 ;  x<32 ; x++){
               out.print("<option value='" + x + "'>" + x + "</option>");
```

```
    }
   %>
  </select>
  <select name="month">
    <option></option>
    <option value="01">January</option>
    <option value="02">Febuary</option>
    <option value="03">March</option>
    <option value="04">April</option>
    <option value="05">May</option>
    <option value="06">June</option>
    <option value="07">July</option>
    <option value="08">August</option>
    <option value="09">September</option>
    <option value="10">October</option>
    <option value="11">November</option>
    <option value="12">December</option>
  </select>
  <select name="year">
    <option></option>
    <%
    for(int y = 1900 ; y <2012 ; y++){
  out.print("<option value='" + y + "'>" + y + "</option>");
    }
    %>
  </select>
</td>
```

```
                </tr>
                <tr>
                  <td></td>
                  <!-- Printing out an a message if the update was successful or there is an error --
>
                  <td style="font-size:12pt; color:dodgerblue;"><%out.println(monthError + "" +
dobUpdate);%></td>
                </tr>
                <tr>
                  <td></td>
                  <!-- Printing out an a message if the update was successful or there is an error --
>
                  <td style="font-size:12pt; color:dodgerblue;"><%out.println(febError);%></td>
                </tr>
                <tr>
                  <td>Change your password</td>
                  <td><input type="password" name="pwd"/></td>
                </tr>
                <tr>
                  <td></td>
                  <!-- Printing out an a message if the update was successful or there is an error --
>
                  <td style="font-size:12pt;
color:dodgerblue;"><%out.println(lengthError);%></td>
                </tr>
                <tr>
                  <td>Confirm the password change</td>
                  <td><input type="password" name="cnpwd"/></td>
                </tr>
```

```
        <tr>
          <td></td>
          <!-- Printing out an a message if the update was successful or there is an error --
>
          <td style="font-size:12pt; color:dodgerblue;"><%out.println(matchError + "" +
passwordUpdate);%></td>
        </tr>
        <tr>
          <td>Change your email</td>
          <td><input type="text" name="email"/></td>
        </tr>
        <tr>
          <td></td>
          <!-- Printing out an a message if the update was successful or there is an error --
>
          <td style="font-size:12pt; color:dodgerblue;"><%out.println(emailError + "" +
emailUpdate);%></td>
        </tr>
        <tr>
          <td>Change your security question</td>
          <td>
            <select name="secq">
              <option></option>
              <option value="pet">What is your first pets name?</option>
              <option value="mum">What is your mums name?</option>
              <option value="teacher">What is your first teachers name?</option>
              <option value="town">What is your home towns name?</option>
              <option value="phone">What is the name of your fav phone?</option>
            </select>
            </td>
        </tr>
        <tr>
            <td></td>
            <!-- Printing out an a message if the update was successful -->
            <td style="font-size:12pt; color:dodgerblue;"><%out.println(secQUpdate);%></td>
        </tr>
        <tr>
            <td>Change the answer</td>
            <td><input type="text" name="seca"/></td>
        </tr>
        <tr>
            <td></td>
            <!-- Printing out an a message if the update was successful or there is an error -->
            <td style="font-size:12pt; color:dodgerblue;"><%out.println(answerError + "" + secAUpdate);%></td>
        </tr>
        <tr>
            <td style="font-size:14pt">
                <input type="submit" value="Change"/>
                <input type="reset" value="Reset"/>
            </td>
            <td colspan="3">
                <a href="login.jsp" style="text-decoration:none; font-size:14pt;">Back to Home page</a>
            </td>
        </tr>
```

```
                </table>

                <hr style="color:dodgerblue" width="550px"/>

                <hr style="color:red" width="550px"/>

              </b>

            </form>

          </td>

        </tr>

        <!-- A simple footer -->

        <tr style="font-weight:bold;font-size:10pt;color:red;">

          <td valign="top">

            <p align="left">

              Tejas Dwarkaram

            </p>

          </td>

          <td valign="top">

            <p align="right">

              NS2012-0340

            </p>

          </td>

        </tr>

      </table>

    </body>

</html>
```

topicspage.jsp

```
<!--
-- Author              Tejas Dwarkaram
-- FileName              topicspage.jsp
-- Date               2012/11/01

-- File Description       JSP File that displays information,
                 comments, and replies to any topic
                 the user selects from the login
                 homepage
-->

<%-- Creating the taglib tag to import the custom tag used for the banner --%>
<%@taglib uri="//WEB-INF/tlds/LogoFormatTag.tld" prefix="ban" %>
<html>

  <head>
    <title>
      <%
      /*
      *creating a string variable to retrieve the active session
      *object that contains a attribute, that signifies if the user
      *logged in or not
       */
      String user = (String)session.getAttribute("user");
      String topic = request.getParameter("topic");
      //getting the topicId of the topic to be displayed
```

```
        String topicId = request.getParameter("topicId");

        out.println("CrossFire&trade;: " + topic + "");

        %>

    </title>

</head>


<body style="background:darkgray">

    <%

    /*

    *checking if the user is logged in, else

    *redirecting to the index page,

    *and stating he needs to login

    */

    if(user == null || user == ""){

            response.sendRedirect("index.jsp?logged=false");

    }

    %>

    <table align="center" style="width:600px;">

        <!-- Using the custom tag that was imported in the taglib tag -->

        <tr>

            <td valign="top" colspan="2">

                <ban:BannerTag/>

            </td>

        </tr>

        <tr>

            <td colspan="2" valign="top" align="center">

                <table width="600px" border="2" style="font-size:14pt; border-color:black;">
```

```
<tr style="color:dodgerblue" >
  <tr>
    <td colspan="4">
      Topic Description
    </td>
    <td>
      Date:
    </td>
    <td>
      <%
      try { // This code block invokes the ForumWebServicePort:checkLogin operation on web service
        //creating a new instance of the implementation of the web service
        ForumService.ForumWebServiceService forumWebServiceService = new ForumService.ForumWebServiceService_Impl();
        //using the implementation of the web service to get the active port used by the web service
        ForumService.ForumWebService forumWebServicePort = forumWebServiceService.getForumWebServicePort();
        //printing out the date the topic was posted
        out.println(forumWebServicePort.getTopicDatePosted(topic));
      %>
    </td>
  </tr>
  <tr>
    <td colspan="7">
      <%
      //printing out the description of the topic selected
      out.println(forumWebServicePort.getTopicDescription(topic));
```

```
                    //catching any exceptions that might occur from the webservice
                    } catch(javax.xml.rpc.ServiceException ex) {
                      System.err.println("ServiceException on topicsPage:" + ex.toString());
                    } catch(java.rmi.RemoteException ex) {
                      System.err.println("RemoteExcepetion on topicsPage :" + ex.toString());
                    } catch(Exception ex) {
                      System.err.println("Unknown Error on topicsPage :" + ex.toString());
                    }
                    %>
                  </td>
                </tr>
              </tr>
            </table>
            <hr/>
            <hr/>
            <big style="font-size:18; colour:gray"><b><u>Comments and Replies to this topic
</u></b></big>
            <hr/>
            <hr/>
          </td>
        </tr>
        <!-- Creaing the HTML table layout to display the comments and replies
           for this topic
         -->
        <tr>
          <td colspan="2" valign="top" align="center">
            <table width="600px" style="font-size:14pt; border-color:black;">
              <tr>
```

```
                    <td>
                      <%
                      /*
                      *printing out the contents of the comments and replies tables
                      *that are associated with the topic selected
                      */
                      try {
                        //creating a new instance of the implementation of the web service
                        ForumService.ForumWebServiceService forumWebServiceService = new
ForumService.ForumWebServiceService_Impl();
                        //using the implementation of the web service to get the active port used by the
web service
                        ForumService.ForumWebService forumWebServicePort =
forumWebServiceService.getForumWebServicePort();
                        //creating a string array to hold the values returned
                        String comments[] = forumWebServicePort.getComments(topic);
                        for(int y = 0; y < comments.length; y++){
                          //creating a new HTML table for each comment returned
                          out.println("<table align='center' width='600px' border='3' style='font-
size:14pt; border-color:black;'>");
                          //Creating an array of the individual values in the elements
                          String[] allComments = comments[y].split(",");
                          int count = 0;
                          String commentId = "";
                          for(int x = 0; x < allComments.length; x++){
                            count++;
                            /*
                            *Performing an if..else statement to check the position of the element
                            *to determine date, user, comment, or the id of the comment
```

```
*/
if(count == 1){
    commentId = allComments[x];
}else if(count == 2){
    //printing out who posted the topic
    out.println("<td valign='top' align=left colspan='2'><font style='color:red'>Comment posted by user : </font><font style='color:blue'>" + allComments[x] + "</font></td>");
}else if(count == 3){
    out.println("<tr>");
    //printing out what the topic is
    out.println("<td colspan='2' align=center valign='top'>" + allComments[x] + "</td>");
    out.println("</tr>");
}else if(count == 4){
    out.println("<tr>");
    //printing out when the topic was posted
    out.println("<td valign='top' align='right' colspan='2'> Date posted : " + allComments[x] + "<hr/></td>");
    out.println("</tr>");
    out.println("<form action='ReplyServlet'><tr><td valign='top' align=right colspan='2' valign='top'><textarea name='replyArea'></textarea><input type='hidden' value='" + commentId + "' name='commentId'/><input type='submit' value='Reply to this'/><hr/></td></tr></form>");
    out.println("</table>");
    //creating a string array to hold the replies returned for this element
    String replies[] = forumWebServicePort.getReplies(commentId);
    for(int e = 0; e < replies.length; e++){
        //Creating an array of the individual values in the elements
        String[] allReplies = replies[y].split(",");
```

```
                    int counter = 0;

                    for(int w = 0; w < allReplies.length; w++){

                        //creating a new row to align to the bottom right of the comment

                        out.println("<tr><td colspan='2'><table align='right' style='font-size:11pt; border-color:black;'>");

                        counter++;

                        if(counter == 1){

                            //printing out who replied

                            out.println("<tr><td valign='top' align='center' colspan='2'><u><font style='color:green'>Reply by : </font><font style='color:aqua'>" + allReplies[w] + "</font></u></td></tr>");

                        }else if(counter == 2){

                            out.println("<tr>");

                            //printing out what they replied

                            out.println("<td colspan='2' valign='top' align='right'>" + allReplies[w] + "</td>");

                            out.println("</tr>");

                        }else if(counter == 3){

                            out.println("<tr>");

                            //printing out when they replied

                            out.println("<td align='right' valign='top'> Date posted : " + allReplies[w] + "</td>");

                            out.println("</tr>");

                        }

                        //closing the html table

                        out.println("</table></td></tr>");

                    }

                }

            }
```

```
                        }
                    }
                    //catching any errors that might occur from the web service
                } catch(javax.xml.rpc.ServiceException ex) {
                    System.err.println("ServiceException found on topics page : " + ex.toString());
                } catch(java.rmi.RemoteException ex) {
                    System.err.println("RemoteException found on topics page : " + ex.toString());
                } catch(Exception ex) {
                    System.err.println("UnknownException found on topics page : " + ex.toString());
                }
                %>
            </td>
        </tr>
    </table>
  </td>
</tr>
<!-- creating the layout for the user to post a new comment to the topic -->
<tr>
  <td colspan="2" valign="top" align="center">
    <!-- Sending the contents of this form to the CommentServlet -->
    <form action="CommentServlet">
      <table align="center" width="600px" style="font-size:14pt; border-color:black;">
        <tr>
          <td>
            Post a comment to this topic :
          </td>
          <td align="right" valign="top">
```

```
            <textarea name="commentArea"></textarea>

          </td>

          <td>

            <input type="hidden" value="<%out.println(topicId);%>" name="topicId"/>

          </td>

        </tr>

        <tr>

          <td colspan="2" align="right">

            <input type="submit" value="Post Comment"/>

          </td>

        </tr>

      </table>

    </form>

  </td>

</tr>

<tr>

  <td>

    <a href="login.jsp">Go Back</a>

  </td>

</tr>

<!-- A simple footer for the page -->

<tr style="font-weight:bold;font-size:10pt;color:red;">

  <td valign="top">

    <p align="left">

      Tejas Dwarkaram

    </p>

  </td>
```

```
        <td valign="top">
          <p align="right">
            NS2012-0340
          </p>
        </td>
      </tr>
    </table>
  </body>
</html>
```

BannerTag.java

```
/*

 * @author Tejas Dwarkaram

 * @version 1.5.0_9

 * @since  18 October 2012, 10:09

 *

 *

 *  fileName    BannerTag.java

 *  description creating a java file to create

 *             a custom tag, that will display

 *             the banner on every jsp page

 */


package ForumService;


import java.io.IOException;

import javax.servlet.jsp.tagext.*;

import javax.servlet.jsp.JspWriter;

import javax.servlet.jsp.JspException;


/**

 *

 * @author  Administrator

 * @version

 */


public class BannerTag extends TagSupport {
    /**Called by the container to invoke this tag.
     * The implementation of this method is provided by the tag library developer,
     * and handles all tag processing, body iteration, etc.
     */
    public int doStartTag() throws JspException {
        try {
            //creating a new JspWriter
            JspWriter out = pageContext.getOut();
            //printing out the html code to display he image
            out.println("<img src=\"both.gif\" width=\"600px\" height=\"100px\"/>");
        } catch(IOException ioe) {
            System.out.println("Client Error -> IOException on FormatTag : " + ioe.getMessage());
        }
        return(EVAL_BODY_INCLUDE);
    }

    public int doEndTag() throws JspException {
        JspWriter out = pageContext.getOut();
        return(EVAL_PAGE);
    }
}
```

CommentServlet.java

```
/*
 * @author Tejas Dwarkaram
 * @version 1.5.0_9
 * @since 08 November 2012, 02:30
 *
 *
 *  fileName    CommentServlet.java
 *  description Creating a servlet to check the
 *              details of a comment, and then
 *              access the webservice and insert
 *              the new comment
 */

package ForumService;

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

/**
 *
 * @author Students
 * @version
 */
public class CommentServlet extends HttpServlet {

   /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
```

```
     * @param request servlet request
     * @param response servlet response
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        //creating a new Printwriter object to write to the web page
        PrintWriter out = response.getWriter();

        //creating a new session object
        HttpSession session = request.getSession();
        //using the session object to retrieve the users name
        String userName = (String)session.getAttribute("user");
        //storing the parameter that holds the text of the comment posted
        String userComment = request.getParameter("commentArea");
        //converting the parameter recieved as the topic id
        int topic = Integer.parseInt(request.getParameter("topicId").trim());
        //creating a new instance of the SimpleDateFormat to format the date
        java.text.SimpleDateFormat sdf = new java.text.SimpleDateFormat("yyyy-MM-dd");
        //getting the current system time and date
        java.util.Date date = new java.util.Date();
        //formatting the current date using the SimpleDateFormat object
        String fullDate = sdf.format(date);

        try { // This code block invokes the ForumWebServicePort:postComments operation on web
service
            //creating a new instance of the implementation of the web service
            ForumService.ForumWebServiceService forumWebServiceService = new
ForumService.ForumWebServiceService_Impl();

            //using the implementation of the web service to get the active port used by the web service

            ForumService.ForumWebService forumWebServicePort =
forumWebServiceService.getForumWebServicePort();

            //sending the correct details to the webservice

            forumWebServicePort.postComments(userName, userComment, fullDate, topic);

            //catching any exceptions that might occur

        } catch(javax.xml.rpc.ServiceException ex) {

            System.out.println("Client Error -> Service Exception on comment servlet : " + ex.toString());

        } catch(java.rmi.RemoteException ex) {

            System.out.println("Client Error -> Remote Exception on comment servlet : " + ex.toString());

        } catch(Exception ex) {

            System.out.println("Client Error -> Unknown Exception on comment servlet : " +
ex.toString());

        }



        //redirecting the page bakc to the login homepage, so the new comment can be seen

        response.sendRedirect("topicspage.jsp");

        //closing the output stream that was created

        out.close();

    }



    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left
to edit the code.">

    /** Handles the HTTP <code>GET</code> method.

     * @param request servlet request

     * @param response servlet response

     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }

    /** Handles the HTTP <code>POST</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }

    /** Returns a short description of the servlet.
     */
    public String getServletInfo() {
        return "Short description";
    }
    // </editor-fold>
}
```

LoginServlet.java

```
/*
 * @author Tejas Dwarkaram
 * @version 1.5.0_9
 * @since  24 October 2012, 10:54
 *
 *
 *  fileName    LoginServlet.java
 *  description Creating a servlet to access the
 *            webservice and check if the login
 *            details are correct
 */

package ForumService;

import java.io.*;

import javax.servlet.*;
import javax.servlet.http.*;

/**
 *
 * @author Students
 * @version
 */
public class LoginServlet extends HttpServlet {

   /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
   * @param request servlet request

   * @param response servlet response

   */

  protected void processRequest(HttpServletRequest request, HttpServletResponse response)

  throws ServletException, IOException {

    response.setContentType("text/html;charset=UTF-8");

    PrintWriter out = response.getWriter();



    boolean correct = false;

    //getting the username parameter from the request object

    String username = request.getParameter("username");

    //getting the password parameter from the request object

    String password = request.getParameter("password");

    try { // This code block invokes the ForumWebServicePort:checkLogin operation on web service

      //creating a new instance of the implementation of the web service

      ForumService.ForumWebServiceService forumWebServiceService = new
ForumService.ForumWebServiceService_Impl();

      //using the implementation of the web service to get the active port used by the web service

      ForumService.ForumWebService forumWebServicePort =
forumWebServiceService.getForumWebServicePort();

      //storing the boolean value returned from the web service on checkin login

      correct = forumWebServicePort.checkLogin(username, password);

      //catching any exceptions if they might occur

    } catch(javax.xml.rpc.ServiceException ex) {

      System.err.println("Client Error -> ServiceException on login page :" + ex.toString());

    } catch(java.rmi.RemoteException ex) {

      System.err.println("Client Error -> RemoteExcepetion on login page :" + ex.toString());

    } catch(Exception ex) {
      System.err.println("Client Error -> UnknownException on login page :" + ex.toString());
    }

    //checking if the login returned true
    if(correct){
      //creating a new session object
      HttpSession session = request.getSession();
      //creating a new attribute to hold that a user is logged in
      session.setAttribute("user", username);
      //redirecting the page to the login homepage
      response.sendRedirect("login.jsp?user=" + username);
    }else{
      //if the login was false, redirecting to the index page
      response.sendRedirect("index.jsp?error=1");
    }
    out.close();
  }

  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
  /** Handles the HTTP <code>GET</code> method.
   * @param request servlet request
   * @param response servlet response
   */
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
  throws ServletException, IOException {
    processRequest(request, response);
  }
    /** Handles the HTTP <code>POST</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }
}
```

ProfileServlet.java

```
/*

 * @author Tejas Dwarkaram

 * @version 1.5.0_9

 * @since  07 November 2012, 09:42

 *

 *

 *  fileName    LoginServlet.java

 *  description Creating a servlet to access the

 *              webservice and check if the login

 *              details are correct

 */


package ForumService;


import java.io.*;

import java.util.regex.Pattern;


import javax.servlet.*;

import javax.servlet.http.*;


/**

 *

 * @author Students

 * @version

 */

public class ProfileServlet extends HttpServlet {
```

```
/** Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
 * @param request servlet request
 * @param response servlet response
 */
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();


    //creating a new session object
    HttpSession session = request.getSession();
    //getting the 'user' attribute from the session object
    String userName = (String)session.getAttribute("user");


    boolean added = false;
    //getting the parameters required for registration and checking if they are empty or not
    String name = request.getParameter("name").trim() == null ? "" :
request.getParameter("name").trim();
    String surname = request.getParameter("surname").trim() == null ? "" :
request.getParameter("surname").trim();
    String year = request.getParameter("year").trim() == null ? "" :
request.getParameter("year").trim();
    String month = request.getParameter("month").trim() == null ? "" :
request.getParameter("month").trim();
    String day = request.getParameter("day").trim() == null ? "" :
request.getParameter("day").trim();
    String pass = request.getParameter("pwd").trim() == null ? "" :
request.getParameter("pwd").trim();
    String pass2 = request.getParameter("cnpwd").trim() == null ? "" :
request.getParameter("cnpwd").trim();
```

```
    String email = request.getParameter("email").trim() == null ? "" :
request.getParameter("email").trim();

    String secQ = request.getParameter("secq").trim() == null ? "" :
request.getParameter("secq").trim();

    String secA = request.getParameter("seca").trim() == null ? "" :
request.getParameter("seca").trim();


    boolean allContents = true;


    //adding the year, month and day variables together

    String fullDate = year + "-" + month + "-" + day;

    //creating an array of all the variables

    String[] allArray = {name, surname, day, month,

    year, pass, pass2, email, secA};

    //creating an array of the months of the year

    String[] months = {"January", "February", "March", "April",

    "May","June","July","August","September",

    "October", "November", "December"};

    //sotring default values to represent the data

    String[] values = {"Name", "Surname", "Day", "Month",

    "Year","Password", "Confirm Password",

    "Email Address",  "Security Answer"};


    String errors = "";

    String updated = "";


    boolean contents = false;


    try { // This code block invokes the ForumWebServicePort:checkLogin operation on web service
```

```
        //creating a new instance of the implementation of the web service
        ForumService.ForumWebServiceService forumWebServiceService = new
ForumService.ForumWebServiceService_Impl();
        //using the implementation of the web service to get the active port used by the web service
        ForumService.ForumWebService forumWebServicePort =
forumWebServiceService.getForumWebServicePort();

        /*
         *Checking if the name variable exists or is empty
         */
        if(name != "" || name != null){
          /*
           *If Statement to check that the name only contains letters
           */
          if(!Pattern.matches("[A-Za-z]+", name)){
            //adding the number to errors found
            errors = errors + "1";
            //setting the boolean variable to false, because of an error
            allContents = false;
          }
          if(allContents){
            forumWebServicePort.editName(name, userName);
            updated = updated + "1";
          }
        }

        /*
         *Checking if the surname variable exists or is empty
```

```
 */
if(surname != "" || surname != null){
   /*
    *If Statement to check that the surname only contains letters
    */
   if(!Pattern.matches("[A-Za-z]+", surname)){
      //adding the number to errors found
      errors = errors + "2";
      //setting the boolean variable to false, because of an error
      allContents = false;
   }
   if(allContents){
      forumWebServicePort.editSurname(surname, userName);
      updated = updated + "2";
   }
}

/*
 *Checking if the date variable exists or is empty
 */
if(fullDate!= "" || fullDate != null){
   /*
    *If statement to check the feb only has less than 29 days
    */
   if(month.equals("02")){
      if(Integer.parseInt(day) > 29){
         //adding the number to errors found
```

```
            errors = errors + "4";
            //setting the boolean variable to false, because of an error
            allContents = false;
         }
      }
      /*
       *If statement to check that certain
       *months only contain 30 days
       */
      if(month.equals("04") || month.equals("06") ||
            month.equals("09") || month.equals("11")){
         if(Integer.parseInt(day) > 30){
            //adding the number to errors found
            errors = errors + "3";
            //setting the boolean variable to false, because of an error
            allContents = false;
         }
      }
      if(allContents){
         forumWebServicePort.editDob(fullDate, userName);
         updated = updated + "3";
      }
   }

   /*
    *Checking if the pass variable exists or is empty
    */
```

```
if(pass != "" || pass != null){
   /*
    *If statement to check if the passwords match eachother
    */
   if(!pass.equals(pass2)){
      //adding the number to errors found
      errors = errors + "6";
      //setting the boolean variable to false, because of an error
      allContents = false;
   }
   /*
    *If statement to check that the length of the password
    *is good
    */
   if(pass.length()<8 || pass.length()>28){
      //adding the number to errors found
      errors = errors + "5";
      //setting the boolean variable to false, because of an error
      allContents = false;
   }
   if(allContents){
      forumWebServicePort.editPassword(pass, userName);
      updated = updated + "4";
   }
}

/*
```

```
 *Checking if the secQ variable exists or is empty
 */
if(secQ != "" || secQ != null){
   forumWebServicePort.editSecQ(secQ, userName);
   updated = updated + "5";
}

/*
 *Checking if the secA variable exists or is empty
 */
if(secA != "" || secA != null){
   /*
    *If statment to check that the security answer conatins valid characters
    */
   if(!Pattern.matches("[\\w]+", secA)){
      //adding the number to errors found
      errors = errors + "8";
      //setting the boolean variable to false, because of an error
      allContents = false;
   }
   if(allContents){
      forumWebServicePort.editSecA(secA, userName);
      updated = updated + "6";
   }
}

/*
```

```
     *Checking if the email variable exists or is empty
     */
    if(email != "" || email != null){
      /*
       *If Statement to check that the email address contains "@" and "."
       */
      if(!Pattern.matches(".+?@.+?\\.{1}.+?", email)){
        //adding the number to errors found
        errors = errors + "7";
        //setting the boolean variable to false, because of an error
        allContents = false;
      }
      if(allContents){
        forumWebServicePort.editEmail(email, userName);
        updated = updated + "7";
      }
    }
  } catch(javax.xml.rpc.ServiceException ex) {
    System.err.println("Client Error -> ServiceException on profile page : " + ex.toString());
  } catch(java.rmi.RemoteException ex) {
    System.err.println("Client Error -> RemoteExcepetion on profile page : " + ex.toString());
  } catch(Exception ex) {
    System.err.println("Client Error -> Unknown Error on profile page : " + ex.toString());
  }

  if(allContents){
    response.sendRedirect("profile.jsp?updated=" + updated);
```

```
        }else{
            response.sendRedirect("profile.jsp?errors=" + errors);
        }
        out.close();
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /** Handles the HTTP <code>GET</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }

    /** Handles the HTTP <code>POST</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }
}
```

RecoveryServlet.java

```
/*

 * @author Tejas Dwarkaram

 * @version 1.5.0_9

 * @since  06 November 2012, 09:06

 *

 *

 *  fileName    RecoveryServlet.java

 *  description Creating a servlet to access the

 *              webservice and help retrieve a lost

 *              or forgotten password

 */


package ForumService;


import com.sun.msv.driver.textui.ReportErrorHandler;

import java.io.*;


import javax.servlet.*;

import javax.servlet.http.*;

import javax.swing.JOptionPane;


/**

 *

 * @author Students

 * @version

 */

public class RecoveryServlet extends HttpServlet {
    /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
     * @param request servlet request
     * @param response servlet response
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        //creating a new session object
        HttpSession session = request.getSession();
        //getting the step attribute from the session
        String number = (String)session.getAttribute("step");

        boolean correct = false;
        //checking which "step" is curently in play
        if(number.equals("1")){
            //getting the email address and name from the request object
            String email = request.getParameter("email");
            String name = request.getParameter("name");

            try { // This code block invokes the ForumWebServicePort:checkLogin operation on web
service
                //creating a new instance of the implementation of the web service
                ForumService.ForumWebServiceService forumWebServiceService = new
ForumService.ForumWebServiceService_Impl();
                //using the implementation of the web service to get the active port used by the web
service
```

```
            ForumService.ForumWebService forumWebServicePort =
forumWebServiceService.getForumWebServicePort();

            //storing the result in a variable

            correct = forumWebServicePort.checkEmail(email, name);

            //catching any exceptions that might occur

        } catch(javax.xml.rpc.ServiceException ex) {

            System.err.println("Client Error -> ServiceException on recovery page :" + ex.toString());

        } catch(java.rmi.RemoteException ex) {

            System.err.println("Client Error -> RemoteExcepetion on recovery page :" + ex.toString());

        } catch(Exception ex) {

            System.err.println("Client Error -> Unknown Error on recovery page :" + ex.toString());

        }


        //if the name and email correspond

        if(correct){

            String question = "";

            try { // This code block invokes the ForumWebServicePort:checkLogin operation on web
service

                //creating a new instance of the implementation of the web service

                ForumService.ForumWebServiceService forumWebServiceService = new
ForumService.ForumWebServiceService_Impl();

                //using the implementation of the web service to get the active port used by the web
service

                ForumService.ForumWebService forumWebServicePort =
forumWebServiceService.getForumWebServicePort();

                //storing the string return in the question variable

                question = forumWebServicePort.getQuestion(email);

            } catch(javax.xml.rpc.ServiceException ex) {

            System.err.println("Client Error -> ServiceException on recovery page :" + ex.toString());
```

```
      } catch(java.rmi.RemoteException ex) {
        System.err.println("Client Error -> RemoteExcepetion on recovery page :" + ex.toString());
      } catch(Exception ex) {
        System.err.println("Client Error -> Unknown Error on recovery page :" + ex.toString());
      }
        //redirecting back to the remember page, which will reset the attribute to step 2
        response.sendRedirect("remember.jsp?step=2&que=" + question);
      }else{
        //redirecting back to the remember page, which will reset the attribute to step 1
        response.sendRedirect("remember.jsp?step=1");
      }
    }else{
      //getting the answer from the request object
      String answer = request.getParameter("answer");
      boolean recovered = false;
      try { // This code block invokes the ForumWebServicePort:checkLogin operation on web
service
        //creating a new instance of the implementation of the web service
        ForumService.ForumWebServiceService forumWebServiceService = new
ForumService.ForumWebServiceService_Impl();
        //using the implementation of the web service to get the active port used by the web
service
        ForumService.ForumWebService forumWebServicePort =
forumWebServiceService.getForumWebServicePort();
        //storing the boolean of whether the answer was correct
        recovered = forumWebServicePort.getAnswer(answer);
      } catch(javax.xml.rpc.ServiceException ex) {
        System.err.println("Client Error -> ServiceException on recovery page :" + ex.toString());
      } catch(java.rmi.RemoteException ex) {
            System.err.println("Client Error -> RemoteExcepetion on recovery page :" + ex.toString());
        } catch(Exception ex) {
            System.err.println("Client Error -> Unknown Error on recovery page :" + ex.toString());
        }

        //checking if true or false
        if(recovered){
            //redirecting to the remember page
            response.sendRedirect("remember.jsp?step=3");
        }else{
            //redirecting to the error page
            response.sendRedirect("remember.jsp?step=1");
        }
    }
    out.close();
  }

  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left
to edit the code.">
  /** Handles the HTTP <code>GET</code> method.
   * @param request servlet request
   * @param response servlet response
   */
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
  throws ServletException, IOException {
    processRequest(request, response);
  }
```

```
    /** Handles the HTTP <code>POST</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }
}
```

RegisterServlet.java

```
/*
 * @author Tejas Dwarkaram

 * @version 1.5.0_9

 * @since  24 October 2012, 03:33

 *

 *

 *  fileName    RegisterServlet.java

 *  description Creating a servlet to check

 *              the values of the register page

 *              and access the web service to

 *              add the new user

 */


package ForumService;


import java.io.*;

import java.util.regex.Pattern;

import javax.servlet.*;

import javax.servlet.http.*;


/**

 *

 * @author Students

 * @version

 */

public class RegisterServlet extends HttpServlet {
```

```
/** Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
 * @param request servlet request
 * @param response servlet response
 */
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
   response.setContentType("text/html;charset=UTF-8");
   PrintWriter out = response.getWriter();

   boolean added = false;
   //getting the parameters required for registration and checking if they are empty or not
   String name = request.getParameter("name").trim();
   String surname = request.getParameter("surname").trim();
   String year = request.getParameter("year").trim();
   String month = request.getParameter("month").trim();
   String day = request.getParameter("day").trim();
   String pass = request.getParameter("pwd").trim();
   String pass2 = request.getParameter("cnpwd").trim();
   String email = request.getParameter("email").trim();
   String secQ = request.getParameter("secq").trim();
   String secA = request.getParameter("seca").trim();

   boolean allContents = true;

   //adding the year, month and day variables together
   String fullDate = year + "-" + month + "-" + day;
   //creating an array of all the variables
```

```
String[] allArray = {name, surname, day, month,

year, pass, pass2, email, secA};

//creating an array of the months of the year

String[] months = {"January", "February", "March", "April",

"May","June","July","August","September",

"October", "November", "December"};

//sotring default values to represent the data

String[] values = {"Name", "Surname", "Day", "Month",

"Year","Password", "Confirm Password",

"Email Address",  "Security Answer"};


String errors = "";


boolean contents = false;

/*

 * For loop to check each value

 * for illegal characters

 */

for(int x=0; x< allArray.length;x++){

   //Regex to check for illegal matches

   if(!Pattern.matches("[.+?@.+?\\.{1}.+?\\w]+", allArray[x])){

      //adding the number to errors found

      errors = errors + "0";

      //setting the boolean variable to false, because of an error

      allContents = false;

      break;

   }
```

```
}
/*
 *If statement to check if the passwords match eachother
 */
if(!pass.equals(pass2)){
   //adding the number to errors found
   errors = errors + "6";
   //setting the boolean variable to false, because of an error
   allContents = false;
}

/*
 *If statement to check that the length of the password
 *is good
 */
if(pass.length()<8 || pass.length()>28){
   //adding the number to errors found
   errors = errors + "5";
   //setting the boolean variable to false, because of an error
   allContents = false;
}

/*
*If Statement to check that the name only contains letters
*/
if(!Pattern.matches("[A-Za-z]+", name)){
   //adding the number to errors found
```

```
    errors = errors + "1";
    //setting the boolean variable to false, because of an error
    allContents = false;
}


/*
 *If Statement to check that the surname only contains letters
 */
if(!Pattern.matches("[A-Za-z]+", surname)){
    //adding the number to errors found
    errors = errors + "2";
    //setting the boolean variable to false, because of an error
    allContents = false;
}


/*
 *If statement to check that certain
 *months only contain 30 days
 */
if(month.equals("04") || month.equals("06") ||
        month.equals("09") || month.equals("11")){
    if(Integer.parseInt(day) > 30){
        //adding the number to errors found
        errors = errors + "3";
        //setting the boolean variable to false, because of an error
        allContents = false;
    }
}



/*
 *If statement to check the feb only has less than 29 days
 */
if(month.equals("02")){
    if(Integer.parseInt(day) > 29){
        //adding the number to errors found
        errors = errors + "4";
        //setting the boolean variable to false, because of an error
        allContents = false;
    }
}



/*
 *If Statement to check that the email address contains "@" and "."
 */
if(!Pattern.matches(".+?@.+?\\.{1}.+?", email)){
    //adding the number to errors found
    errors = errors + "7";
    //setting the boolean variable to false, because of an error
    allContents = false;
}
```

```
        /*
         *If statment to check that the security answer conatins valid characters
         */
        if(!Pattern.matches("[\\w]+", secA)){
            //adding the number to errors found
            errors = errors + "8";
            //setting the boolean variable to false, because of an error
            allContents = false;
        }
        if(allContents){
            try { // This code block invokes the ForumWebServicePort:checkLogin operation on web
service
                //creating a new instance of the implementation of the web service
                ForumService.ForumWebServiceService forumWebServiceService = new
ForumService.ForumWebServiceService_Impl();
                //using the implementation of the web service to get the active port used by the web
service
                ForumService.ForumWebService forumWebServicePort =
forumWebServiceService.getForumWebServicePort();
                //storing the result in the boolean variable
                added = forumWebServicePort.addUser(name, surname, fullDate.trim(), pass, email, secQ,
secA);
                if(added){
                    //redirecting to the new user page
                    response.sendRedirect("newUser.jsp");
                }else{
                    //redirecting back to the register page
                    response.sendRedirect("register.jsp?exist=1");
                }
                //catching any possible exceptions
            } catch(javax.xml.rpc.ServiceException ex) {
                System.err.println("Client Error -> ServiceException on register page : " + ex.toString());
            } catch(java.rmi.RemoteException ex) {
                System.err.println("Client Error -> RemoteExcepetion on register page : " + ex.toString());
            } catch(Exception ex) {
                System.err.println("Client Error -> UnknownException on register page : " + ex.toString());
            }

        }else{
            //redirecting to the register page with errors
            response.sendRedirect("register.jsp?errors=" + errors);
        }
        out.close();
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /** Handles the HTTP <code>GET</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }
```

```
    /** Handles the HTTP <code>POST</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }
}
```

ReplyServlet.java

```
/*
 * @author Tejas Dwarkaram
 * @version 1.5.0_9
 * @since  08 November 2012, 03:51
 *
 *
 *  fileName    ReplyServlet.java
 *  description Creating a servlet to check the
 *              details of a reply, and then
 *              access the webservice and insert
 *              the new reply
 */

package ForumService;

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

/**
 *
 * @author Students
 * @version
 */
public class ReplyServlet extends HttpServlet {

   /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
```

```
     * @param request servlet request
     * @param response servlet response
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        //creating a new PrintWriter object to write to the web page
        PrintWriter out = response.getWriter();

        //creating a new session object
        HttpSession session = request.getSession();
        //using the session object to retrieve the users name
        String userName = (String)session.getAttribute("user");
        //storing the parameter that holds the text of the reply posted
        String replyText = request.getParameter("replyArea");
        //converting the parameter recieved as the comment id
        int commentId = Integer.parseInt(request.getParameter("commentId").trim());
        //creating a new instance of the SimpleDateFormat to format the date
        java.text.SimpleDateFormat sdf = new java.text.SimpleDateFormat("yyyy-MM-dd");
        //getting the current system time and date
        java.util.Date date = new java.util.Date();
        //formatting the current date using the SimpleDateFormat object
        String fullDate = sdf.format(date);

        try { // This code block invokes the ForumWebServicePort:postReplies operation on web service
            //creating a new instance of the implementation of the web service
            ForumService.ForumWebServiceService forumWebServiceService = new
ForumService.ForumWebServiceService_Impl();
```

```
            //using the implementation of the web service to get the active port used by the web service
            ForumService.ForumWebService forumWebServicePort = forumWebServiceService.getForumWebServicePort();
            //sending the correct details to the webservice
            forumWebServicePort.postReplies(userName, replyText, fullDate, commentId);
        } catch(javax.xml.rpc.ServiceException ex) {
            System.out.println("Client Error -> Service Exception on reply servlet : " + ex.toString());
        } catch(java.rmi.RemoteException ex) {
            System.out.println("Client Error -> Remote Exception on reply servlet : " + ex.toString());
        } catch(Exception ex) {
            System.out.println("Client Error -> Unknown Exception on reply servlet : " + ex.toString());
        }

        //redirecting the page bakc to the login homepage, so the new reply can be seen
        response.sendRedirect("topicspage.jsp");
        //closing the output stream that was created
        out.close();
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /** Handles the HTTP <code>GET</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }

    /** Handles the HTTP <code>POST</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }

    /** Returns a short description of the servlet.
     */
    public String getServletInfo() {
        return "Short description";
    }
    // </editor-fold>
}
```

Create_Database_forum_db.sql

```
/********************************************************************
*       Filename: Create_Database_forum_db
            *
*       Author: Tejas Dwarkaram                                     *
*       Date: 20 July 2012
                  *
*       Description: Creating the database for the forum website,   *
*                             as well as adding the tables
            *
*******************************************************************/
USE master
GO

IF EXISTS(SELECT name FROM master.dbo.sysdatabases
            WHERE name = 'forum_db')
            DROP DATABASE forum_db
            GO

--Creating the database
CREATE DATABASE forum_db
ON PRIMARY
(
      NAME = 'forum_data',
      FILENAME = 'C:\forum_db.mdf',
      SIZE = 10,
      MAXSIZE = UNLIMITED,
      FILEGROWTH = 2
)
--creating the log file for the database
LOG ON
(

      NAME = 'forum_log',
      FILENAME = 'C:\forum_db.ldf',
      SIZE =10,
      MAXSIZE = UNLIMITED,
      FILEGROWTH = 2
)
GO

USE forum_db
GO

--creating all of the tables for the database
CREATE TABLE topics
(
      topic_id int NOT NULL IDENTITY(1, 1),
      topic_name VARCHAR(500) NOT NULL,
      topic_decs VARCHAR(1000) NOT NULL,
      topic_user VARCHAR(40) NOT NULL,
      topic_date VARCHAR(12) NOT NULL,
      CONSTRAINT prim_topics_id PRIMARY KEY(topic_id)
)
GO

CREATE TABLE comment
(
      comment_id int NOT NULL IDENTITY(1,1),
      users_name VARCHAR(40) NOT NULL,
	user_text VARCHAR(1000) NOT NULL,
	comment_date VARCHAR(12) NOT NULL,
	topic_id int NOT NULL,
	CONSTRAINT for_comment_id FOREIGN KEY(topic_id) REFERENCES
topics(topic_id),
	CONSTRAINT prim_comment_id PRIMARY KEY(comment_id)
)
GO

CREATE TABLE replies
(
	users_name VARCHAR(40) NOT NULL,
	user_reply VARCHAR(1000) NOT NULL,
	reply_date VARCHAR(12) NOT NULL,
	comment_id int NOT NULL,
	CONSTRAINT for_reply_id FOREIGN KEY(comment_id) REFERENCES
comment(comment_id)
)

CREATE TABLE users
(
	users_id int NOT NULL IDENTITY(1,1),
	users_name VARCHAR(15) NOT NULL,
	users_surname VARCHAR(40) NOT NULL,
	users_dateOfBirth DATE NOT NULL,
	users_password VARCHAR(20) NOT NULL,
	users_email VARCHAR(30) NOT NULL,
	user_secQ VARCHAR(50) NOT NULL,
	user_secA VARCHAR(20) NOT NULL,
	CONSTRAINT prim_user_id PRIMARY KEY(users_id)
)
GO
```