

# Term Project: Option 1

By: Morgan Smith, Ruddy Leon, Steven Szachara, Ben Schneider

# Authentication

```
29
30
31
32 <a href="http://www.cuse.com"> Watch Now </a>
33 <div class="container">
34 <head>
35 <script>
36 function validateForm() {
37     var x = document.forms["emailval"]["uemail"].value;
38     var atpos = x.indexOf("@syr");
39     var dotpos = x.lastIndexOf(".");
40     if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length) {
41         alert("Not a valid e-mail address");
42         event.preventDefault()
43         return false;
44     }
45 }
46 }
```

- Code for authentication
- Requests credentials to compare what is provided to a file in our database of authorized users information on the local operating system within the authentication server
- If the credentials match, the process is completed and the user is granted authorization for access
- This also covers Post

# CSS

```
1  /* The Modal (background) */
2  .modal {
3      display: none; /* Hidden by default */
4      position: fixed; /* Stay in place */
5      z-index: 1; /* Sit on top */
6      left: 0;
7      top: 0;
8      width: 100%; /* Full width */
9      height: 100%; /* Full height */
10     overflow: auto; /* Enable scroll if needed */
11     background-color: rgb(0,0,0); /* Fallback color */
12     background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
13     padding-top: 60px;
14 }
15
16 .div1 {
17     text-align: center;
18     margin: auto;
19     font-size: 35px;
20     width: 80%;
21 }
```

```
19     font-size: 35px;
20     width: 80%;
21 }
22 /* Modal Content/Box */
23 .modal-content {
24     background-color: #fefefe;
25     margin: 5px auto; /* 15% from the top and centered */
26     width: 80%; /* Could be more or less, depending on screen size */
27 }
28
29 /* The Close Button */
30 .close {
31     /* Position it in the top right corner outside of the modal */
32     position: absolute;
33     right: 25px;
34     top: 0;
35     color: #000;
36     font-size: 35px;
37     font-weight: bold;
38 }
39
40 /* Close button on hover */
41 .close:hover,
42 .close:focus {
43     color: red;
```

# Index

```
1 <html>
2
3
4 <link rel="stylesheet" type="text/css" href="style.css" />
5 <head>
6     <div class="div1">
7         <head>
8             Welcome to SU Sport's Open Forum
9         </head>
10    </div>
11
12    <!--Button to open the modal login form -->
13
14    <!--<button onclick="document.getElementById('id01').style.display='block'" -->
15    <!-- The Modal
16    <div id="id01" class="modal">
17        <span onclick="document.getElementById('id01').style.display='block'"
18        class="close" title="Close Modal">&times;</span>
19    -->
20    <!-- Modal Content -->
21    <!--
22    <form class="modal-content animate" action="page2.html">
23    -->
24    <body>
25        <div class="imgcontainer">
26
27            
28
29        </div>
30
```

```
16 <div id="id01" class="modal">
17     <span onclick="document.getElementById('id01').style.display='block'"
18     class="close" title="Close Modal">&times;</span>
19 -->
20 <!-- Modal Content -->
21 <!--
22 <form class="modal-content animate" action="page2.html">
23 -->
24 <body>
25     <div class="imgcontainer">
26
27         
28     </div>
29
30
31
32 <a href="http://www.cuse.com"> Watch Now </a>
33 <div class="container">
34 <head>
35 <script>
36 function validateForm() {
37     var x = document.forms["emailval"]["uemail"].value;
38     var atpos = x.indexOf("@");
39     var dotpos = x.lastIndexOf(".");
40     if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length) {
41         alert("Not a valid e-mail address");
42         event.preventDefault();
43         return false;
44     }
45 }
```

# JHTTP

```
import java.io.*;
import java.net.*;
import java.util.concurrent.*;
import java.util.logging.*;

public class JHTTP {

    private static final Logger logger = Logger.getLogger(
        JHTTP.class.getCanonicalName());
    private static final int NUM_THREADS = 50;
    private static final String INDEX_FILE = "index.htm";

    private final File rootDirectory;
    private final int port;

    public JHTTP(File rootDirectory, int port) throws IOException {

        if (!rootDirectory.isDirectory()) {
            throw new IOException(rootDirectory
                + " does not exist as a directory");
        }
        this.rootDirectory = rootDirectory;
        this.port = port;
    }

    public void start() throws IOException {
        ExecutorService pool = Executors.newFixedThreadPool(NUM_THREADS);
        try (ServerSocket server = new ServerSocket(port)) {
            logger.info("Accepting connections on port " + server.getLocalPort());
            logger.info("Document Root: " + rootDirectory);
```

```
            while (true) {
                try {
                    Socket request = server.accept();
                    Timer timer = new Timer(4000);
                    timer.start();
                    Runnable r = new RequestProcessor(
                        rootDirectory, INDEX_FILE, request);
                    pool.submit(r);
                    timer.stop();
                } catch (IOException ex) {
                    logger.log(Level.WARNING, "Error accepting connection", ex);
                }
            }
        }
    }

    public static void main(String[] args) {

        // get the Document root
        File docroot;
        try {
            docroot = new File(args[0]);
        } catch (ArrayIndexOutOfBoundsException ex) {
            System.out.println("Usage: java JHTTP docroot port");
            return;
        }
    }
```

# JHTTP cont.

```
// set the port to listen on
int port;
try {

    port = Integer.parseInt(args[1]);
    if (port < 0 || port > 65535) port = 80;
} catch (RuntimeException ex) {
    port = 80;
}

try {
    JHTTP webserver = new JHTTP(docroot, port);
    webserver.start();
} catch (IOException ex) {
    logger.log(Level.SEVERE, "Server could not start", ex);
}
}
```

- This covers HTTP
- This covers Multi-Threading
- This covers logging
- This covers GET

# Request Processor

```
import java.io.*;
import java.net.*;
import java.nio.file.Files;
import java.util.*;
import java.util.logging.*;

public class RequestProcessor implements Runnable {

    private final static Logger logger = Logger.getLogger(
        RequestProcessor.class.getCanonicalName());

    private File rootDirectory;
    private String indexFileName = "index.html";
    private Socket connection;

    public RequestProcessor(File rootDirectory,
        String indexFileName, Socket connection) {

        if (rootDirectory.isFile()) {
            throw new IllegalArgumentException(
                "rootDirectory must be a directory, not a file");
        }
        try {
            rootDirectory = rootDirectory.getCanonicalFile();
        } catch (IOException ex) {
        }
        this.rootDirectory = rootDirectory;

        if (indexFileName != null) this.indexFileName = indexFileName;
        this.connection = connection;
    }
}
```

```
@Override
public void run() {
    // for security checks
    String root = rootDirectory.getPath();
    try {
        OutputStream raw = new BufferedOutputStream(
            connection.getOutputStream()
        );
        Writer out = new OutputStreamWriter(raw);
        Reader in = new InputStreamReader(
            new BufferedInputStream(
                connection.getInputStream()
            ), "US-ASCII"
        );
        StringBuilder requestLine = new StringBuilder();
        while (true) {
            int c = in.read();
            if (c == '\r' || c == '\n') break;
            requestLine.append((char) c);
        }

        String get = requestLine.toString();

        logger.info(connection.getRemoteSocketAddress() + " " + get);
    }
```



# Request Processor cont.

```
String[] tokens = get.split("\\s+");
String method = tokens[0];
String version = "";
if (method.equals("GET")) {
    String fileName = tokens[1];
    if (fileName.endsWith("/")) fileName += indexFileName;
    String contentType =
        URLConnection.getFileNameMap().getContentTypeFor(fileName);
    if (tokens.length > 2) {
        version = tokens[2];
    }

    File theFile = new File(rootDirectory,
        fileName.substring(1, fileName.length()));

    if (theFile.canRead())
        // Don't let clients outside the document root
        && theFile.getCanonicalPath().startsWith(root)) {
        byte[] theData = Files.readAllBytes(theFile.toPath());
        if (version.startsWith("HTTP/")) { // send a MIME header
            sendHeader(out, "HTTP/1.0 200 OK", contentType, theData.length);
        }

        // send the file; it may be an image or other binary data
        // so use the underlying output stream
        // instead of the writer
        raw.write(theData);
        raw.flush();
    }
}
```

```
} else { // can't find the file
    String body = new StringBuilder("<HTML>\r\n")
        .append("<HEAD><TITLE>File Not Found</TITLE>\r\n")
        .append("</HEAD>\r\n")
        .append("<BODY>")
        .append("<H1>HTTP Error 404: File Not Found</H1>\r\n")
        .append("</BODY></HTML>\r\n").toString();
    if (version.startsWith("HTTP/")) { // send a MIME header
        sendHeader(out, "HTTP/1.0 404 File Not Found",
            "text/html; charset=utf-8", body.length());
    }
    out.write(body);
    out.flush();
}

} else { // method does not equal "GET"
    String body = new StringBuilder("<HTML>\r\n")
        .append("<HEAD><TITLE>Not Implemented</TITLE>\r\n")
        .append("</HEAD>\r\n")
        .append("<BODY>")
        .append("<H1>HTTP Error 501: Not Implemented</H1>\r\n")
        .append("</BODY></HTML>\r\n").toString();
    if (version.startsWith("HTTP/")) { // send a MIME header
        sendHeader(out, "HTTP/1.0 501 Not Implemented",
            "text/html; charset=utf-8", body.length());
    }
    out.write(body);
    out.flush();
}

} catch (IOException ex) {
    logger.log(Level.WARNING,
        "Error talking to " + connection.getRemoteSocketAddress(), ex);
}
```



# Request Processor cont.

```
    } finally {  
        try {  
            connection.close();  
        }  
        catch (IOException ex) {}  
    }  
}
```

```
private void sendHeader(Writer out, String responseCode,  
    String contentType, int length)  
    throws IOException {  
    out.write(responseCode + "\r\n");  
    Date now = new Date();  
    out.write("Date: " + now + "\r\n");  
    out.write("Server: JHTTP 2.0\r\n");  
    out.write("Content-length: " + length + "\r\n");  
    out.write("Content-type: " + contentType + "\r\n\r\n");  
    out.flush();  
}
```

# Timer

```
class Timer extends Thread {
    /** Rate at which timer is checked */

    protected int m_rate = 100;

    /** Length of timeout */
    private int m_length;

    /** Time elapsed */
    private int m_elapsed;

    /**
     * Creates a timer of a specified length
     * @param length
     * Length of time before timeout occurs
     */

    public Timer ( int length )
    {
        // Assign to member variable

        m_length = length;

        // Set time elapsed

        m_elapsed = 0;
    }
}
```

```
    }

    /** Resets the timer back to zero */
    public synchronized void reset()
    {
        m_elapsed = 0;
    }

    /** Performs timer specific code */
    public void run()
    {
        // Keep looping
        for (;;)
        {
            // Put the timer to sleep

            try
            {
                Thread.sleep(m_rate);
            }
        }
    }
}
```

# Timer

```
}  
  
catch (InterruptedException ioe)  
{  
    continue;  
}  
  
// Use 'synchronized' to prevent conflicts  
  
synchronized ( this )  
{  
  
    // Increment time remaining  
  
    m_elapsed += m_rate;  
  
    // Check to see if the time has been exceeded  
  
    if (m_elapsed > m_length)  
    {  
  
        // Trigger a timeout  
  
        timeout();  
  
    }  
}
```

```
public void timeout()  
{  
    System.err.println ("Network timeout occurred.... terminating");  
    System.exit(1);  
}  
}
```

# Our Idea

Twitter API Host Page for SU

---

# Log On Page

SU Twitter Hub. For all things SU.



[Go to the Syracuse Homepage!](#)

School Email   ☒ Remember me


# Twitter Page

## Twitter Feed

### Tweets by @dailyorange

**The Daily Orange** @dailyorange  
Lindsay Eastwood was supposed to be on blood thinners for life. [bit.ly/2gqxIFD](http://bit.ly/2gqxIFD)

“  
I never thought we'd be doing this again.  
DAVID EASTWOOD  
Lindsay Eastwood's father on watching her play hockey



2h

Embed

View on Twitter

### Tweets by @SyracuseUNews

**Syracuse Univ. News** @SyracuseUNews  
Check out our new look at [news.syr.edu](http://news.syr.edu)  
[@SyracuseU](https://twitter.com/SyracuseU)



08 Dec

Embed

View on Twitter

### Tweets by @SUCampus

**SU Campus** @SUCampus  
Seniors! The #SyracuseU Yearbook is now available for order: [ow.ly/ooUI306YvqQ](http://ow.ly/ooUI306YvqQ)

SU Campus Retweeted

**Syracuse Alumni** @SUALums  
GEORGETOWN TAILGATE: We're kicking off a series of alumni tailgates NEXT Sat. before @CUSE\_MBB takes on Georgetown! [cc.syr.edu/bballtailgates...](http://cc.syr.edu/bballtailgates...)



7h

Embed

View on Twitter

### Tweets by @DOsports

**Daily Orange Sports** @DOsports  
Amba Etta-Tawo named to All-America Third Team, becomes first SU All-American since 2001: [bit.ly/2hm8P8b](http://bit.ly/2hm8P8b)

### Tweets by @universityunion

**UNIVERSITY UNION** @universityunion  
Obsessed with @HamiltonMusical? How about @chancerapper? We're jamming to "Dear Theodosia" on the new Hamilton mixtape! 🌟 #MusicMonday

### Tweets by @iSchoolSU

**Syracuse iSchool** @iSchoolSU  
The SU Climate Assessment Planning Committee is holding another conversation session tomorrow at 4:00 PM in Schine. [bit.ly/2hb5g4V](http://bit.ly/2hb5g4V)



# QUESTIONS?

