Design document
for CSC/ECE 517 Fall 2002
Semester Project
Security & Visibility for PG

Authors:
Jianjun Huo   jhuo@unity.ncsu.edu
Lilian Seow   lseow@us.ibm.com
Mack Steadman msteadman1@triad.rr.com
Tiejun Li     tli5@unity.ncsu.edu
1. Introduction ......................................................................................................................... 3

1.1 Purpose .......................................................................................................................... 3

1.2 Scope .................................................................................................................................. 3

1.2.1 Read-protect submissions ......................................................................................... 3

1.2.2 System Files Viewing .............................................................................................. 3

1.2.3 Administrator to view reviews and reviews of reviews ........................................... 3

1.2.4 A logout function ...................................................................................................... 3

1.3 Background ...................................................................................................................... 4

1.3.1 Read-protect submissions ......................................................................................... 4

1.3.2 System Files Viewing .............................................................................................. 4

1.3.3 Administrator to view reviews and reviews of reviews ............................................ 4

1.3.4 A logout function ...................................................................................................... 5

1.4 Documentation ................................................................................................................ 5

1.5 Standards ......................................................................................................................... 5

1.6 Other documents .............................................................................................................. 5

2. Product Requirements ....................................................................................................... 5

2.1 Read-protect submissions ............................................................................................. 6

2.2 System Files Viewing ..................................................................................................... 6

2.3 Administrator to view reviews and reviews of reviews ................................................ 7

2.4 Logout function .............................................................................................................. 7

3. Testing Requirements ........................................................................................................ 7

3.1 Read protect .................................................................................................................... 7

3.2 System Files Viewing ...................................................................................................... 8

3.3 Administrator to view reviews and reviews of reviews ................................................ 9

3.4 Logout function .............................................................................................................. 9

4. Design .................................................................................................................................. 10

4.1 Class Diagram .................................................................................................................. 10

4.1.1 Read protect ............................................................................................................. 10

4.1.2 System Files Viewing .............................................................................................. 11

4.1.3 Administrator to view reviews and reviews of reviews ........................................... 12

4.1.4 A logout function ...................................................................................................... 13

4.2 Use cases ........................................................................................................................ 13

4.2.1 Read protect ............................................................................................................. 13

4.2.2 System File Viewing .............................................................................................. 14

4.2.3 Administrator to view reviews and reviews of reviews ............................................ 16

4.2.4 A logout function ...................................................................................................... 16

Figure 1 Class diagram for Read-Protect ............................................................................... 10
Figure 2 Class diagram for System File Viewing ................................................................... 11
Figure 3 Class diagram for Administrator to reviews ......................................................... 12
Figure 4 Class diagram for Logout function .......................................................................... 13
1. Introduction

1.1 Purpose

This project is to implement the requirements stated in the “Security & Visibility for PG” project description:
  v Read-protect every submissions, only authorized user can see
  v Provide a graphical interface for administrator to view all files
  v Provide a way that administrator can view reviews and reviews of reviews
  v Provide a logout function

1.2 Scope

This design unit covers all modifications/improvements related to the project description listed in section 1.1. This design won’t cover any existing PG system features or functionalities which are not changed here. Readers may go to https://pg.csc.ncsu.edu/, to find online documentation or contact the site administrator for help.

1.2.1 Read-protect submissions

The scope of this enhancement shall be limited to providing a mechanism that will only protect the contents under directory /pg/pg_data/public/. It assumes the directory scheme under /pg/pg_data/public/ is consistent and the data it requires to perform the privilege check are available according to the existing PG system implementation.

1.2.2 System Files Viewing

The scope of this enhancement shall be limited to providing a “standalone” user interface that will be initiated from the “Link Page” of the Peer Grading system. Where as there may exist opportunities to initiate this new functionality as options from within other components of the Peer Grading system, the integration of this functionality into these components will not be within this defined scope.

1.2.3 Administrator to view reviews and reviews of reviews

The scope of this enhancement shall be limited to providing a system administrator certain privileges, including: viewing the submission of the reviews, the reviews of the reviews, and so on. The functionality will be supported from GUI for administrators to view all files.

1.2.4 A logout function

The scope of this enhancement shall be limited to providing a “logout” link only on the PG system home page.
1.3 Background

Readers of this document should be familiar with the concept of Peer Grading (PG), and the PG system that is currently running on NCSU campus. PG white paper can be found at [https://pg.csc.ncsu.edu/PG_efg.pdf](https://pg.csc.ncsu.edu/PG_efg.pdf).

The Peer Grading system is an object-oriented system written in Java. The system is used by Dr. Gehringer to provide students a repository for research documents ("submissions") that are to be evaluated by the peers of the submitter.

During this evaluation, peers are provided an opportunity to review a given submission, provide comments to it and to register grades based on the given evaluation.

Each semester students are given the opportunity via a “semester project” to enhance the system itself using Java. So lies the underlying purpose of this project.

1.3.1 Read-protect submissions

Currently, submissions are stored in the directories under /pg/pg_data/public/, any user connected to the Internet is able use a Internet browser to browse the directory hierarchy and retrieve any submissions available there.

To limit unexpected activities before the submissions are officially published to the public, a mechanism is need to restrict users’ accessibility to unpublished submissions.

1.3.2 System Files Viewing

To provide this functionality to submitters and reviews, there exist several administrative functions the must be performed previous to the actual usage. These range anywhere from defining the assignment to be performed to assigning peer reviewers for potential submitters.

Administrative functions are supported by several internal files, each containing information necessary for ongoing system availability (i.e. passwords, logons, assignments, etc.). Presently the only means of reviewing file support hierarchy is by going directly into the command line functionality of the Linux server. Where as this is quite doable, it is also very burdensome.

A user-friendlier means of reviewing the file support hierarchy is desired.

1.3.3 Administrator to view reviews and reviews of reviews

To provide this functionality to the administrators, there exist several administrative functions the must be performed. It includes viewing the reviews and the reviews of reviews.
In order to limit these functionalities only to system administrators, we need to provide a mechanism to restrict the access to corresponding files containing the reviews and the reviews of reviews.

1.3.4 A logout function

In current PG system, there is no logout function available. In order to change the role, a user has to close all Internet browser windows, and then reopen the browser, re-connect to the PG web site, and re-login. This seems hassle to people when performing system administrative tasks.

To allow the user be able to change role without terminating the Internet browser, a logout function is needed.

1.4 Documentation

Existing PG system administration documentation can be found at http://pg.csc.ncsu.edu/doc/. Presently, there is no complete PG system user’s manual available. It needs to be generated someday in the future (when required and appropriate), articulating the operation, administration and maintenance procedures of the existing PG system. Of course, the modifications/improvements implemented in this design document shall be reflected in that user’s manual as well.

1.5 Standards

In this design unit, the standards that we are referring/using are all contained in the existing PG system. Those standards include:

- HTML
- Servlet
- Tomcat
- Java

1.6 Other documents

Readers may find other useful information at:

- http://java.sun.com

2. Product Requirements

In this project, we are required to apply these changes to the current in-production PG system:

- An option to read-protect every submission – so no other user can read it. (Read-protection needs to be turned on, if it is desired for submission to be on the Web
while they are under review, and turned off when projects are “published” to the Web.

- A way for an administrator to view all files from a graphical interface that has keyboard shortcuts. (E.g., the administrator should be able to walk the directory tree from /public on down.)
- A way for an administrator to view reviews and reviews of reviews associated with a submission without having to log in as the student who submitted it.
- A logout function for PG (to allow logging in as a different user without closing the browser). It is acceptable to provide the option to log out only on the PG home page.

2.1 Read-protect submissions

In the current PG system, users are able to retrieve any submissions by browsing the directory hierarchy under https://pg.csc.ncsu.edu/pg/pg_data/public/, without the need to log in first.

The tasks in this design unit are to:
R1.1 Protect the directory hierarchy from being browsed by users, which means users will not able to freely use Internet browsers to traverse the directory hierarchy.

R1.2 After submission but before being published, documents can only be viewed by: 1) owner; 2) reviewers; 3) administrator

R1.3 After being published, documents are open to the public.

2.2 System Files Viewing

In the current PG system, administrator has no GUI tools to traverse the file directory hierarchy.

The tasks in this design unit are to:
R2.1 - Provide the user of means of specifying the beginning point in the hierarchy traversal process.

R2.2 Provide the user a means to specifically identify where in the directory hierarchy they are currently located.

R2.3 Provide the user a means to hit hot key(s) that will direct them to
a) a specific point in the directory hierarchy.
b) The next point in the directory hierarchy.
c) The previous point in the directory hierarchy

R2.4 Provide the user a means of spawning the DirectoryTraverser object from the Link Page web page
2.3 Administrator to view reviews and reviews of reviews

In the current PG system, administrator has no methods to view the reviews and the reviews of reviews.

The tasks in this design unit are to:
R3.1 - Provide the administrators of means of viewing the reviews of some specific submissions.

R3.2 - Provide the administrators of means of viewing the reviews of reviews which are submitted by the reviewers.

2.4 Logout function

In the current PG system, users don’t have a Logout option on the web. So, if the user wants to change his/her role, he/she has to close all browser window first and then reopen Internet browser in order to be able to log in again.

The tasks in this design unit are to:
R4.1 Provide a logout hyperlink at the PG homepage, which will invalidate current user session when being clicked, and redirect the user to the login page

R4.2 Check that every page should have a link back to the PG homepage, if there is not, provide one.

3. Testing Requirements

3.1 Read protect

T1.1 Ensure that the owner can retrieve the submissions.

T1.2 Ensure the reviewers are able to retrieve the submissions.

T1.3 Ensure the administrator is able to retrieve the submissions.

T1.4 Ensure that before submissions are published, users other than the owner, reviewers, and administrator, can not retrieve the submissions.

T1.5 Ensure that after submissions are published, users other than the owner, reviewers, and administrator, can retrieve the submissions.

T1.6 Ensure that invalid or malicious user input are detected and rejected.
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1.1 Protect the directory hierarchy from being browsed by users, which means users will not able to freely use Internet browsers to traverse the directory hierarchy.</td>
<td>T1.4</td>
</tr>
<tr>
<td>R1.2 After submission but before being published, documents can only be viewed by: 1) owner; 2) reviewers; 3) administrator</td>
<td>T1.1, T1.2, T1.3</td>
</tr>
<tr>
<td>R1.3 After being published, documents are open to the public</td>
<td>T1.5</td>
</tr>
</tbody>
</table>

3.2 System Files Viewing

T2.1 Ensure that the user can be notified of the current directory that they are in.

T2.2 Ensure that the user is able to see the files encountered during traversal

T2.3 Ensure that the path is highlighted as the user traverses the directory hierarchy

T2.4 Ensure that the directory traversal functions successfully starts up on a request from the Link Page.

T2.5 Ensure that the user can successfully terminate the traversal process function

T2.6 Ensure that the user can start the traversal process at the desired directory.

T2.7 Ensure that user is able to traverse the desired directory in a forward direction.

T2.8 Ensure that the user is able to traverse the desired directory in a backward direction

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2.1 - Provide the user of means of specifying the beginning point in the hierarchy traversal process.</td>
<td>T2.6</td>
</tr>
<tr>
<td>R2.2 Provide the user a means to specifically identify where in the directory hierarchy they are currently located</td>
<td>T2.1</td>
</tr>
<tr>
<td>R2.3 Provide the user a means to hit hot key(s) that will direct them to a) a specific point in the directory</td>
<td>T2.2, T2.6, T2.7, T2.8</td>
</tr>
<tr>
<td>Requirement</td>
<td>Test</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>T2.4, T2.5</td>
<td></td>
</tr>
</tbody>
</table>

### 3.3 Administrator to view reviews and reviews of reviews

T3.1 Ensure the administrator to be able to retrieve the submissions of the reviews.
T3.2 Ensure the administrator to be able to retrieve the submissions of the reviews of reviews.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>T3.1</td>
<td></td>
</tr>
<tr>
<td>T3.2</td>
<td></td>
</tr>
</tbody>
</table>

### 3.4 Logout function

T1.1 Ensure that the logout link on the PG system homepage works.
T1.2 Ensure that all other pages has a hyperlink pointing to the home page.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1.1</td>
<td></td>
</tr>
<tr>
<td>T1.2</td>
<td></td>
</tr>
</tbody>
</table>
4. Design

4.1 Class Diagram

4.1.1 Read protect

Figure 1 Class diagram for Read-Protect

In the above class diagram, class FileGuard is the new class that we added to protect submissions. Class LoginPage, Student, and Assignment, are the classes that have direct interaction with class FileGuard. Submission document are not real Java classes in the PG system, they are files comprises of submissions. Directory under public are not Java classes either, they are directories where submission documents reside.

Class FileGuard is a Java Servlet. When it is deployed into the PG system, the URL /pg/pg_data/public/ will be mapped to this Servlet. This means, when the user is using Internet browser to try to fetch document under directory /pg/pg_data/public/, he/she won’t be able to direct reach the file as before. Instead, FileGuard servlet is invoked. The file location after /pg/pg_data/public is passed to FileGuard as part of the HttpRequest. FileGuard checks the user’s role, user’s id, his assignments, his review assignments and the submission status to determine whether the access should be granted. If access should be granted, FileGuard retrieves the file and wrap the file content into a
MIME response, and then send it to the browser. If access is denied, FileGuard will display an access denied page.

Since FileGuard is designed to serve the access not only to HTML files but also other file types, this feature is transparent to all existing PG implementations. There is no additional changes need to be made to other existing PG Java classes.

FileGuard is designed to be robust enough to reject any invalid and malicious request from the Internet browser.

### 4.1.2 System Files Viewing

![Class diagram for System File Viewing](image)

Figure 2 Class diagram for System File Viewing

A new object is needed to be defined in the system call DirectoryTraverser. This object will provide functionality to traverse a desired directory hierarchy.
4.1.3 Administrator to view reviews and reviews of reviews

Here we can design a java class AdminView that can be used only by the system administrators. In this class, we will provide several methods such as reading the files of reviewers’ submissions. When the administrator want to see these submissions, we will initialize this object and use the methods it provides to retrieve the corresponding files from certain directories.
4.1.4 A logout function

A button will be added to the PG homepage (Login.html) and a Java Servlet called Logout will also be added to the PG system. The Logout Servlet will be invoked to invalidate the session and expire the cache when the user clicks on the logout button. After the logout button is clicked, a popup dialog will come up to confirm that the user has logged out from the system. To log back in, the user can click on the proceed button (currently existing in the PG system).

![Figure 4 Class diagram for Logout function](image)

4.2 Use cases

4.2.1 Read protect

R1.1 Protect the directory hierarchy from being browsed by users, which means users will not able to freely use Internet browsers to traverse the directory hierarchy.

Input:  User input submission location into the Internet Explorer’s address field

Description: The full file path at which the submission resides according to the HTTP specification. The user hasn’t logged in yet, or logged in but not the owner, reviewers or administrator of this submission. Submission hasn’t been published yet.

- Type of Input: URL
- Source of Input: Internet Explorer

Processing: Check the user privilege, and denies the access.

Outputs: An HTML page shows access denied information.
R1.2 After submission but before being published, documents can only be viewed by: 1) owner; 2) reviewers; 3) administrator

Input: User input submission location into the Internet Explorer’s address field
Description: The full file path at which the submission resides according to the HTTP specification. The user has logged in as owner, reviewers or administrator of this submission. Submission hasn’t been published yet.

Type of Input: URL
Source of Input: Internet Explorer

Processing: Check the user privilege, and access is granted

Outputs: The submission asked.

R1.3 After being published, documents are open to the public.

Input: User input submission location into the Internet Explorer’s address field
Description: The full file path at which the submission resides according to the HTTP specification. The user has not logged in.
Submission has been published.

Type of Input: URL
Source of Input: Internet Explorer

Processing: access is granted

Outputs: The submission asked.

4.2.2 System File Viewing

R2.1 - Provide the user of means of specifying the beginning point in the hierarchy traversal process.

Input: Directory designation
Description: The directory at which the traversal should start
Type of Input: String
Source of Input: User interface

Processing: Attempt to find the directory on the server

Outputs: If the directory is found on the server display the directory and all of the Sub-directories down the hierarchy in the traversal window
Display all files contained in the given directory in the file list window.

If directory is not found on the server, display message to user:
“Your starting point is not a directory”
R2.2 Provide the user a means to specifically identify where in the directory hierarchy they are currently located.

Input: Action Event
Description: Button pressed event
Type of Input: Action Event
Source of Input: User interface

Processing: Locate the current directory you are in

Outputs: Show the current directory you are in the current directory text field.

R2.3 Provide the user a means to hit hot key(s) that will direct them to
a) The previous point in the directory hierarchy.

Input: Up Arrow
Description: Up Arrow key is pressed
Type of Input: Action Event
Source of Input: User interface

Processing: Locate the parent directory

Outputs: Show in the file list windows all files that are contained in
Parent directory of the directory which was listed at the time the up arrow was hit.

b) The previous point in the directory hierarchy

Input: Down Arrow
Description: Down Arrow key is pressed
Type of Input: Action Event
Source of Input: User interface

Processing: Locate the next child directory

Outputs: Show in the file list windows all files that are contained in
Child directory of the directory which was listed at the time the Down arrow was hit.

R2.4 Provide the user a means of spawning the DirectoryTraverser object from the Link Page web page

Input: Action Event
Description: Directory Traversal Button is pressed on Link Page
Type of Input: Action Event
Source of Input: User interface
Processing: Create new DirectoryTraversal Object

Outputs: Directory traversal Gui Interface

4.2.3 Administrator to view reviews and reviews of reviews

R3.1 - Provide the administrators of means of viewing the reviews of some specific submissions.

Input: Assignment list, then the student ID (submission) list
Description: The directory at which the traversal should start
Type of Input: Action Event
Source of Input: User interface for the administrators

Processing: Attempt to find the review files on the server

Outputs: If the review files are found on the server display all the review files in the file list window.

If in this directory there are no review files on the server, display message to the administrator:
“Currently there are no review files”

R3.2 - Provide the administrators of means of viewing the reviews of reviews that are submitted by the reviewers.

Input: Assignment list, then the student ID (submission) list
Description: The directory at which the traversal should start
Type of Input: Action Event
Source of Input: User interface for the administrators

Processing: Attempt to find the files of reviews of reviews on the server

Outputs: If the reviews of review files are found on the server display all the files in the file list window.

If there are no reviews of review files on the server, display message to the administrator:
“Currently there are no reviews of review files”

4.2.4 A logout function

R4.1 Provide a logout hyperlink at the PG homepage, which will invalidate current user session when being clicked, and redirect the user to the login page
Input: User clicks the logout hyperlink on the home page.
Description: user previously logged in.
Type of Input: Action Event
Source of Input: Internet browser

Processing: User has been logged out

Outputs: A HTML page shows the user has been logged out. (if the user revisit
the home page and click the “proceed” button, system will ask the user to log in)