Pat Keller
Charles Missi

Project Definition – 1.0
CS 498

Introduction
The proposed project is the CS Gradebook that is being developed for the Computer Science department at Southern Illinois University at Carbondale. This system is to be used by students for the submission of work and viewing class grades. It will also be used by each course's Teaching Assistant (TA) to collect submitted work and publish student grades. The deliverables for this project are source code, technical documentation, and user documentation.

What is the CS Gradebook?
The CS Gradebook is a program designed to combine and overhaul both the current CS Submit and CS Gradebook programs. The program will allow for online student submission of work and grade viewing and allow for course TAs to create assignments and keep course grade books online.

Benefits
The benefit of this new program is that students will be able to submit work through a web application instead of logging in to a remote system to submit work. This program will also allow students to view their grades through a web application from each CS course that they are currently taking. This program will also simplify the process that TAs must use to collect all submitted work and keep track of course grades.

What Problem Does the CS Gradebook Solve?
The Computer Science department needs a way for students to be able to submit class work remotely because the CS computer lab cannot be open 24 hours a day 7 days a week. The current method of submission deletes any previous submissions before copying the specified files to the course directory and does not provide any indication of success once the copying action is complete. There is also no way to see what files have been previously submitted. With the current method of submission, TAs must run a copy command for each of the students who submitted any files for an assignment. This is a tedious process that can sometimes have errors.

The CS department also wants to allow students to view their grades from CS classes through a web application. Some of the CS courses currently post grades online, but a student can only view one course's grades at a time and must change to a different course's website in order to view that course's grades.

How does the CS Gradebook Solve the Problem?
The CS Gradebook will be a web application with an intuitive user interface that will allow for easy submission of work. It will allow students to make submissions from either their Linux home directory or their local hard drive. All of the submissions that a student has submitted previously will also be saved. Students will be able to upload files to their home directory and download files from their home directory. Students will also be able to view all of their CS department grades from the web application. TAs will be able to create submissions with due dates. TAs will also be able to select all of the files that were submitted at once and copy them to one folder for grading. TAs will also be able to enter and store all of their course grades with the application.

Who will use the Gradebook?

The main users for the Gradebook will be the students and the teaching assistants. The students will be using this software for submitting/viewing assignments and viewing grades; whereas the teaching assistants will be inserting grades for each student and creating assignment submissions.

Use Cases

There are seven total use cases involved with this software. These interactions are specific to each of the possible options that are available with the software.

Use Case 1: Submit from Linux Home Directory

Name	**UC-1: Submit from Linux Home Directory**
Summary	A student will be allowed to submit a homework assignment from his/her Linux home directory.
Rationale	A student must be able to submit an assignment.
Users	Student
Preconditions	Student is logged into the grade book.
Basic Course of Events	1) User tells the software to go to the submit screen. 2) User locates the file to be submitted. 3) File is uploaded into the software. 4) User submits the file.
Alternative Paths	1) User decides not to submit the file. User is at the same screen as in Step 1 2) User decides to submit multiple files; steps 2 through 4 are repeated.
Post conditions	The file is submitted to the appropriate sources.

Use Case 2: Submit from local hard drive

Name	**UC-2: Submit from local hard drive**
Summary	A student will be allowed to submit assignments via local hard drive.
Rationale	Students must be able to submit assignments that are located on their local hard drive.
Users	Student
Preconditions	Student is logged into the grade book.

Basic Course of Events	1) User tells the software to go to the 'Submit' screen. 2) User locates the file on the hard drive. 3) File is uploaded into the software. 4) User submits the file.
Alternative Paths	1) User decides not to submit the file. User is at the same screen as in Step 1 2) User decides to submit multiple files; steps 2 through 4 are repeated.
Post conditions	The file is submitted to the appropriate sources.

Use Case 3: View all previous submissions

Name	**UC-3: View all previous submissions**
Summary	A student will be allowed to view all previous submissions.
Rationale	Lets the student know which files they have already submitted so they can confirm if a file has been submitted.
Users	Student
Preconditions	Student is logged into the grade book and has submitted files previously.
Basic Course of Events	1) User tells the software to go to the 'Check Previous Submissions' screen. 2) User selects which class he wants to view the submissions for. 3) User selects which assignment he wants to view the submissions for. 4) All submissions are shown.
Alternative Paths	1) A user could pick a different course for viewing the submission, the user would jump to step 2. 2) A user could pick a different assignment for viewing the submission, the user would jump to step 3. 3) A user could abandon this task by exiting the program or returning to a different screen.
Post conditions	All previous submissions are shown.

Use Case 4: View all grades

Name	**UC-4: View all grades**
Summary	A student will be allowed to view all of their grades in each computer science course.
Rationale	To allow the student to make sure all grades are reported accurately.
Users	Student
Preconditions	Student is logged into the grade book and is taking the class he/she wants to view.
Basic Course of Events	1) User tells the software that he/she wants to view grades. 2) Software takes student to 'View Grades' screen. 3) User tells software which course to view grades for. 4) The user's grades are shown.
Alternative Paths	1) User decides to view a different course, therefore; must go back to step 2.
Post conditions	All of the user's grades for the course are viewable.

Use Case 5: Create Submissions

Name	**UC-5: Create Submissions**
Summary	A TA(Teaching Assistant) will be able to create a submission for an assignment
Rationale	The TA must have this ability to create a submission for the course the TA is assigned to.
Users	Teaching Assistant
Preconditions	TA is logged into the grade book.
Basic Course of Events	1) User tells software to take him to the 'Create Submission' screen. 2) Software displays editable data fields for both assignment & due date. 3) User enters the assignment name. 4) User enters the due date. 5) User creates the assignment.
Alternative Paths	1) User does not want to create a submission. The user thus exits this screen to return to any previous screen the user wants.

Post conditions	The new submission is uploaded for users to submit their assignments.

Use Case 6: View Submissions

Name	**UC-6: View Submissions**
Summary	A TA will be allowed to view all submissions.
Rationale	In order to verify who has submitted their assignment, this use case must be implemented.
Users	Teaching Assistant
Preconditions	User is logged into the grade book.
Basic Course of Events	1) User tells the software to go to the 'View Submission' screen. 2) User chooses which assignment to view all submissions for. 3) User is shown all assignments that have been uploaded by students.
Alternative Paths	1) User wants to view a different class for submission. Go to step 2 to view different classes.
Post conditions	All files for that assignment are available for view.

Use Case 7: Create Grades

Name	**UC-7: Create Grades**
Summary	A TA will be allowed to enter grades for a student
Rationale	The TA must be able to enter grades into the grade book so other students are able to view their grades.
Users	Teaching Assistant.
Preconditions	TA is logged into the grade book.
Basic Course of Events	1) User tells software to go to 'Create Grade' screen. 2) User selects which class to create a grade for. 3) User selects which student to enter a grade for. 4) User enters the grade for the particular student. 5) User accepts the changes to the Gradebook & the grade is shown for that particular student and

	their assignment.
Alternative Paths	1) User decides to abort the process of entering grade and returns to any previous screen.
Post conditions	The grade for that student is updated and reflected in the Gradebook.

Functional Requirement

Functional requirements describe the main features that the software will provide for the user.

Functional Requirement 1: Gradebook Access

Name:	**FR-1: Gradebook Access**
Summary	The user must be able to access and log into the Gradebook from any location with internet access.
Rationale	The users must be able to access the Gradebook for this to have purpose.
Requirements	When the user wants to access the Gradebook, they navigate with their web browser to the Gradebook URL and log into the system. If a user is unable to log into the system, the application will display an error message letting them know what problem occurred.
Reference	N/A

Functional Requirement 2: Display User's Home Directory

Name:	**FR-2: Display User's Home Directory**
Summary	The user must be able to view their home directory upon successfully logging into the Gradebook.
Rationale	Users will upload files to their home directory, download files from their home directory, and submit files from their home directory so they should be able to view their home directory.

Requirements	When a user has successfully logged into the Gradebook, their home directory is shown in their web browser.
Reference	UC1: Submit from Linux Home Directory

Functional Requirement 3: Submit an Assignment

Name:	**FR-3: Submit an Assignment**
Summary	The user must be able to submit files for their assignments.
Rationale	Users need to complete their coursework.
Requirements	The user must be logged into the Gradebook and select the class and assignment which they wish to submit one or more files to. The user then specifies what file or files they wish to submit and the files are copied to that assignment's folder in the course home directory.
Reference	UC1: Submit from Linux Home Directory. UC2: Submit from Local Hard Drive

Functional Requirement 4: Upload/Download to or from Home Directory

Name:	**FR-4: Display User's Home Directory**
Summary	Users must be able to upload files to and download files from their home directories.
Rationale	Users need to be able to see the files that are stored in their home directory.
Requirements	The user must first be logged into the Gradebook. Then, the user's home directory will show in the application.
Reference	N/A

Functional Requirement 5: Save all Submissions

Name:	**FR-5: Save all Submissions**
Summary	All submissions should be saved and never over-written.
Rationale	Sometimes a user will want to see what changes they have made since their previous submission.
Requirements	Whenever a user is making a submission for an assignment in which they already have a submission(s), the application will copy the previous files to another directory for archiving and then copy the new file(s) to that submission's directory.
Reference	UC1: Submit from Linux Home Directory. UC2: Submit from Local Hard Drive. UC3: View all Previous Submissions

Functional Requirement 6: Copy Files to a Local Computer

Name:	**FR-6: Copy Files to a Local Computer**
Summary	A course TA will copy all files that were submitted for an assignment to their local computer for grading.
Rationale	The TAs need to view all of the files so that they can be graded.
Requirements	The TA must be logged into the Gradebook with course account. They will see a list of all the submissions. They will then select download files, and select all of the files to download to their computer.
Reference	UC6: View Submissions

Functional Requirement 7: View Grades

Name:	**FR-7: View Grades**
Summary	Student users will be viewing their CS course grades.
Rationale	Students need to be able to keep track of their scores in one location.
Requirements	The student must be logged into the Gradebook and select the "View Grades" option. Then, the user chooses the course they wish to view and his/her grades for that course are shown.
Reference	UC4: View all Grades. UC7: Create Grades

Functional Requirement 8: Enter Grades

Name:	**FR-8: Enter Grades**
Summary	TAs will be entering and storing grades with the application.
Rationale	Grades only need to be stored in one location so that students can view them.
Requirements	The course TA must be logged into the course account. They select "Create Grades" and select the student and assignment and enter a grade.
Reference	UC7: Create Grades

Nonfunctional Requirements

These requirements will be used to judge the operation of the software, not the specific behavior of the software.

NF-1: Inherent User Interface

Name:	**NF-1: Inherent User Interface**
Summary	Create an interface which allows all users to feel comfortable & at ease in using the software.
Rationale	Making sure the interface looks sleek is imperative since we are redesigning a Gradebook that many students are already familiar with. Creating a sloppy interface will cause users to want to revert back to the old system.
Requirements	Intuitive yet simple design to let users have a pleasant experience using the software. Without a good looking GUI, no person will ever want to use this.
Reference	UC-1: Submit from Linux Home Directory. UC-2: Submit from local hard drive.UC-5: Create Submissions. UC-7: Create Grades. FR-1: Gradebook Access. FR-2: Display User Home Directory. FR-6: View Gradebook.

NF-2: User Restricted to View Only their Home Directory

Name:	**NF2: User Restricted to View Only their Home Directory**
Summary	Users will only be able to see their home directory and will not be able to navigate to any other location.
Rationale	System needs to be secure so that users are not able to steal other's work or view other's files.
Requirements	The permissions of the system will be set in such a way that a logged in user can only view his/her home directory.
Reference	UC-1: Submit from Linux Home Directory. FR-1: Gradebook Access. FR-2: Display User Home Directory.

Technical Specifications

The software will be running on a Linux server, in which the actual software will be available on the website that will be available to all Computer Science majors. Students and TA's will be able to access this website on any operating system platform as long as an internet connection is available. Much of the coding will be accomplished using PHP to create the web application software.