Requirements Specification

Requested by:  Dr. Scott Hunter
Professor of Computer Science
Computer Science Department of Siena College

Ms. Jami Cotler
Professor of Computer Science
Computer Science Department of Siena College

The Electronic Spreadsheet
Automated Teaching Assistant

Pear Software

Prepared by:  Matt Restivo – Team Leader
Colin Cubinski
Daniel Nakhla
James Rocco
Justin Valentini

October 23rd, 2006
Table of Contents:

Section 1.1: Product Overview and Summary ................................................................. 2
Section 1.2: Development, Operating and Maintenance Environments ...................... 2
Section 1.3: External Interfaces and Data Flows .......................................................... 3
Section 1.4: Functional Requirements ........................................................................ 23
Section 1.5: Performance Requirements .................................................................... 24
Section 1.6: Exception Handling .................................................................................. 24
Section 1.7: Early Subsets and Implementation Priorities .......................................... 24
Section 1.8: Foreseeable Modifications and Enhancements ....................................... 24
Section 1.9: Acceptance Criteria ................................................................................. 25
Section 1.10: Testing Requirements ............................................................................ 27
Section 1.11: Design Hints and Guidelines ................................................................. 27
Section 1.12: Cross Reference Index ......................................................................... 27

Appendices:

Section 2.1: Glossary of Terms .................................................................................... 28
Section 2.2: Gantt Chart ............................................................................................. 29
Section 1.1: Product Overview and Summary

A computer automated spreadsheet file assessor and grader will take a major burden off of busy college professors that do not have time to grade a large amount of assignments. Our clients, Dr. Scott Hunter and Ms. Jami Cotler have the problem of too many papers to grade and not enough time to do it. Pear Software can offer them a solution, one which will cut down on the tedium of grading Microsoft Excel Spreadsheet assignments for the Computer Science 010 class at Siena College, which a large amount of students are required to take. The E-SATA program will grade lab and pre-lab excel files using a master answer key.

Section 1.2: Development, Operating and Maintenance Environments

The greater part of development will take place in the facilities provided for by Dr. Lederman, which is the Software Engineering Lab in Roger Bacon at Siena College. The operating of our software will take place in the Computer Science labs of Roger Bacon, in a Windows XP environment. Maintenance environments are not finalized at this stage, however they are assumed to be the similar machines as ones currently in the Roger Bacon Computer Labs.
Section 1.3: External Interfaces and Data Flows

This prototype is a screen of the initial interface any user will see when they access e-SATA. If they are already registered with the system, they will simply type in their user name and password to gain access to the system. If they are not a registered user with the system, they will have the option to create a new account from this screen as well. If someone forgets their password, they can click on the Forgot Password button and it will be e-mailed to them.
This is the screen where a new user can set up their account. The Course and Section drop down menus are specified user-friendly so that students who are new to college are not confused.
This will email the user the password to the email in which they had originally registered.
This prototype is the student user screen. Here they will have the ability to tab through certain screens to access a variety of features, including the ability to view grades on all previous assignments, submit new assignments to be graded, and to view a general feedback report on their assignments, as well as how their classmates did, and where common problems occurred. They also have the option to log out at any time, while any tab is selected. The currently selected tab allows them to see their grades on previously submitted lab reports.
This prototype is the student user screen with the second tab selected. On this menu, they have the ability to view their previously submitted pre-lab reports, as well as re-submit a pre-lab report as long as it is not past the due date. The file will need to be re-uploaded to the system with changes before it will be allowed to be re-submitted. The student user can also tab over to another menu from here.
This prototype is the student user screen, with the submit file option selected. Here, the student can choose a file from a directory to submit to the system for automated grading. The file will need to be in Microsoft Excel format for it to be accepted. Student’s can also re-submit pre-labs here, overwriting the previous one if they wish to re-submit it.
This prototype is the Feedback generator screen. Student’s will have the option to choose a lab or pre-lab, and view the feedback report that was generated upon its automated grading, showing their mistakes. This report will also contain general information from all other lab sessions, designating things such as common mistakes and problems.
This screen shows what it will look like when a user views their feedback. It will flag the bad cells red and pop up with a yellow box to explain the errors.
This prototype is the initial screen a lab instructor will see upon logging in to the system. This user will have two options, a grading tab and a feedback tab. The currently selected tab which is grading, allows the lab instructor to edit all lab and pre-lab assignment grades here, as well as keep track of a student’s attendance, which is automatically taken when they access the system. They can also override attendance if one of their students attends another lab, or they themselves have a guest student in their session. The user will select a student by name, course, or session, to access their information.
This prototype is the feedback screen for a lab instructor. They can select a student by name, course, or section, and view a feedback report of the student’s mistakes on their assignments. The lab instructor also has the right to view a common error feedback report, allowing them to see where most mistakes occurred, and how frequently.
This prototype is the initial screen a lecture instructor will see upon logging in to the system. This user will have two options, a grading tab and a feedback tab. The currently selected tab which is grading, allows the lab instructor to keep track of a student’s grade for the lab and pre-labs. The lecture instructor has the ability to view any student in any lab that is enrolled in their course. The user will select a student by name, course, or session, to access their information.
This prototype is the feedback screen for a lecture instructor. They can select a student by name, course, section, or session, and view a feedback report of the student’s mistakes on their assignments. The lecture instructor also has the right to view a common error feedback report, allowing them to see where most mistakes occurred, and how frequently.
This prototype is the initial screen a course administrator will see when they log into the system. This user will have the right to view each and every student’s lab grades, feedback reports, as well as the common error feedback report. The currently selected tab is for viewing grades.
This prototype is the feedback report screen for the course administrator. Here, they can view feedback reports for any individual student taking the course, as well as view the common error feedback report.
This prototype is the upload answer key screen, exclusively used by the course administrator. Here, the administrator will have the ability to upload spreadsheet answer keys to e-SATA, which will be used to grade student pre-lab and lab reports which is specified by the radio buttons. It will also generate a feedback report for an individual student, and an overall common error feedback report for general viewing.
Level 3 Diagram
Web Interface
Section 1.4: Functional Requirements

Use-Cases for e-SATA:

I) Student:
   • Make submissions of their pre-lab and lab files.
   • Resubmit the pre-lab before the lab session after feedback is received.
   • The feedback screen will provide comments to incorrect cells to show mistakes.
   • View their lab grades.
   • If a student misses a lab and attends another to make it up, the software will be able to recognize the discrepancy and resume normal function with a lab instructor override.

II) Lab Instructor:
   • Lab Instructors will be able to view the profile of any student in a lab session.
   • A profile encompassing the grades and attendance of a student.
   • Viewing the profile signifies the ability to view and edit the students’ grade.
   • e-SATA will grade the pre-lab and lab, and then the lab instructors will scan over the e-SATA assessment and adjust anything that was marked incorrectly (i.e. formulas), or assigning partial credit as necessary.
   • Get feedback on what errors or mistakes made by students were the most frequent.
   • View feedback from other lab sessions.
   • Keep attendance using the application and perform overrides for guest students making up a lab.

III) Lecture Instructor:
   • View their students’ grades
   • View statistics and feedback from all lab sessions.

IV) Course Administrator:
   • View all grades from all lab sections.
   • Upload answer keys to e-SATA for automated grading of pre-labs and labs.

Narrative:

e-SATA will be a web-based application that will allow the user to login and perform specific tasks as per their designation as either a student, lab instructor, lecture instructor, or course administrator. e-SATA will communicate directly with a database, being able to submit and retrieve information at the request of the user, assuming the user has the privileges to access that material. Functions include submission of assignments for automated grading, the ability to view or edit grades and attendance, and to view feedback on graded assignments.
Section 1.5: Performance Requirements

e-SATA’s web based submission and grade/attendance viewing functions will be available for use on Windows, Macintosh, and Linux platforms. This will be designed to be optimally viewed at a 1024 x 768 resolution on the following browsers: Internet Explorer 6, Firefox 1.5, Netscape 8.12, and Safari 2.04. The grading environment will be run on a Windows machine running XP, with the Microsoft Office 2003 Suite.

Section 1.6: Exception Handling

If a user improperly used the software, their will be numerous resolves to stop any type of major problem from occurring.

• Uploading an incorrect file type for the use specified will result in an error message and a prompt to upload a correct file type. (Spreadsheet, answer key, etc.)
• If a file transfer is stopped pre-completion, it will not count as a submission and the user will have another chance to re-submit.
• If the user fails to supply the correct login information, they will be notified and have a chance to fix their error.
• If a user forgets their password, the system will reset it and supply that information to the user’s school e-mail account.

Section 1.7: Early Subsets and Implementation Priorities

The most important aspects of the software include:

• A friendly user interface with easy accessibility to all clients.
• Students will be able to submit their assignments quickly with no confusing or distracting elements.
• Lab instructors will be able to quickly scan over graded files and make corrections.

Section 1.8: Foreseeable Modifications and Enhancements

Possible modifications to the software may include several of the following:

• The ability to grade other types of lab assignments besides Microsoft Excel Spreadsheets.
• A full grade storage system that will allow lecture instructors to store all class grades, not being restrained to just lab grades.
• An improved, detailed grading algorithm they may have more functionality towards grading complex solutions, like cell formulas.
Section 1.9: Acceptance Criteria

E-SATA will contain:

1. Graphical User Interface
2. Grading Engine
3. System Database

E-SATA’s GUI will be broken up into two parts:

1. The Web Interface
2. The Grading System Software

The Web Interface will handle the interaction of the four types of users to the system itself. The Grading System Software will only be used within the program itself. It will be used to grade the user submitted labs in comparison with the uploaded answer key.

The Excel Grading System will have four different types of users, each with their own unique privileges and purposes for the use of e-SATA.

1. Administrator
2. Lab Instructor
3. Lecture Instructor
4. Student

The Course Administrator will be able to view all the grades from all the sections. The Course Administrator will also have full access rights to edit and upload answer keys used by e-SATA to grade the pre-labs and labs.

Lab Instructors will be able to view the profile of any student in a lab session. A profile encompasses the grades and attendance of a student. Viewing the profile signifies the ability to view and edit the students’ grade. e-SATA will grade the pre-lab and lab, and then the lab instructors will scan over the e-SATA assessment and adjust anything that was marked incorrectly (i.e. formulas), or assigning partial credit as necessary. Lab Instructors will also be able to get feedback on what errors or mistakes made by students have the highest percentage. Feedback from other lab sessions will also be available to Lab Instructors. Lab Instructors will be able to keep attendance using the application and perform overrides.
Lecture Instructors will have the right to view their students’ grades and view statistics and feedback from all lab sessions. The feedback includes what errors or mistakes made by students have the highest percentage.

Students will be able to make submissions of their pre-lab and labs files. After submitting their pre-lab they will also have the right to resubmit the pre-lab if they want to make changes before the lab session after feedback is received. The feedback screen will provide comments to incorrect cells to show mistakes; if a mistake is not limited to one cell there will be one dedicated cell to hold all of the notes on these mistakes. If a mistake is limited to only one cell a comment will be added to incorrect cells directly. Students will also have the ability to view their own lab grades. If a student misses a lab and attends another to make it up, the software will be able to recognize the discrepancy and resume normal function with a lab instructor override.

The System Database will be broken up into four separate tables:

1. User Table
2. Assignment Table
3. Key Table
4. Results Table

The User Table will contain a list of all of the users involved with the system. This includes Administrators, Lecture Instructors, Lab Instructors, and Students. This table will contain the privileges assigned to each type of user and will provide them with their corresponding abilities in a dynamically generated Web Interface. The User Table will also be used with the Grading System Software to ensure that the user of the software is of the grader class and will only allow them access to their lab assignments as well as keep track of attendance.

The Assignment Table will contain all assignments submitted by the students. These assignments will not be graded and will be made available for complete access by the Lecturer, Lab instructor and will also be available for the respective student to view.

The Key Table will contain all of the keys submitted by the Administrator. These keys will be used per assignment to grade them. The Grading Engine will make comparisons between the keys, their corresponding assignments, and criteria established by the Grader to formulate results.

The Results Table will contain all of the statistical results and the grades of the students. It will contain both feedback and percentage of the lab which is correct. This information will be available to both the Lab Instructor and Lecture Instructor.
Section 1.10: Testing Requirements

e-SATA will undergo a very detailed, rigorous, and precise testing procedure by Pear Software team members, as well as outside members to make it a fair process. All functionality, from administrator’s key submission down to student file submission will all be thoroughly tested until they meet the criteria specified by the team. Security of files, access speed, and user friendliness will also be high priorities in the testing procedure.

Section 1.11: Design Hints and Guidelines

A student user will submit his assignment through the web application of e-SATA. This assignment will be graded according to the submitted answer key by the course administrator, reviewed by a lab instructor, and feedback will be returned to the student on his work, as well as a grade.

Section 1.12: Cross Reference Index

This section will not be available until the next step of our project as designated by the waterfall model. The preliminary design phase will have a more complete and accurate compilation of data flow diagrams, which are necessary for the completion of this section.
### Section 2.1: Timeline

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project Timeline</td>
<td>65 days</td>
</tr>
<tr>
<td>2</td>
<td>Team Formation</td>
<td>1 day</td>
</tr>
<tr>
<td>3</td>
<td>Client Meetings</td>
<td>55.1 days</td>
</tr>
<tr>
<td>18</td>
<td>Team Meeting</td>
<td>56.23 days</td>
</tr>
<tr>
<td>41</td>
<td>Software Plan</td>
<td>7 days</td>
</tr>
<tr>
<td>42</td>
<td>Software Plan Documentation</td>
<td>6 days</td>
</tr>
<tr>
<td>43</td>
<td>Software Plan Presentation</td>
<td>1 day</td>
</tr>
<tr>
<td>44</td>
<td>Software Specification</td>
<td>29 days</td>
</tr>
<tr>
<td>45</td>
<td>Software Specification Documentation</td>
<td>26 days</td>
</tr>
<tr>
<td>46</td>
<td>Software Specification Presentation</td>
<td>1 day</td>
</tr>
<tr>
<td>47</td>
<td>Preliminary Design</td>
<td>30 days</td>
</tr>
<tr>
<td>48</td>
<td>Preliminary Design Documentation</td>
<td>26 days</td>
</tr>
<tr>
<td>49</td>
<td>Preliminary Design Presentation</td>
<td>1 day</td>
</tr>
</tbody>
</table>

The image also includes a Gantt chart with timelines for various project tasks and milestones, indicating progress and scheduled dates.
Section 2.2: Glossary of Terms

**Code** - The symbolic arrangement of data or instructions in a computer program or the set of such instructions.

**Database** - A collection of data arranged for ease and speed of search and retrieval.

**DBMS (Database Management System)** - Software that controls the organization, storage, retrieval, security and integrity of data in a database.

**Gantt Chart** - A chart that depicts progress in relation to time, often used in planning and tracking a project.

**HTML (Hyper Text Markup Language)** – A markup language used to structure text and multimedia documents and to set up hypertext links between documents, used extensively on the World Wide Web.

**Internet** - An interconnected system of networks that connects computers around the world via the TCP/IP protocol.

**Linear Sequential Model / Classic Waterfall Model** – A systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing, and support.

**PC (Personal Computer)** - Another name for a microcomputer designed for use by a single user.

**PHP (PHP Hypertext Preprocessor)** – A server-side, cross-platform, HTML-embedded scripting language used to create dynamic web pages. PHP is open source software.

**Software** - Written programs or procedures or rules and associated documentation pertaining to the operation of a computer system and that are stored in read/write memory.

**SQL (Structured Query Language)** - Pronounced "sequel", it is a language that provides an interface to relational database systems.

**TCP/IP (Transmission Control Protocol / Internet Protocol)** - the suite of communications protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being Transmission Control Protocol (TCP) and Internet Protocol (IP).