Doornail Technologies, Inc.

System Requirement Specifications for the Grade Book System

CS3300 Software Engineering

16.04.2012

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Release Date | Responsible Party | Major Changes |
| 1.0 | 12.03.2012 |  | Release for design |
| 2.0 | 16.04.2012 |  | Re-release for design |

Table of Contents

[1.0 Introduction 3](#_Toc319178975)

[1.1 Purpose 3](#_Toc319178976)

[1.2 Scope 3](#_Toc319178977)

[1.3 Definitions, Acronyms, and Abbreviations 3](#_Toc319178978)

[2.0 General Description 3](#_Toc319178979)

[2.1 Product Perspective 3](#_Toc319178980)

[2.2 Product Function Summary 3](#_Toc319178981)

[2.3 User Characteristics 4](#_Toc319178982)

[2.4 General Constraints 4](#_Toc319178983)

[2.5 Assumptions and Dependencies 4](#_Toc319178984)

[3.0 Specific Requirements 4](#_Toc319178985)

[3.1 Functional Requirements 4](#_Toc319178986)

[3.2 External Interface Requirements 5](#_Toc319178987)

[3.3 Use Case Descriptions 5](#_Toc319178988)

[3.4 Performance Requirements 5](#_Toc319178989)

[3.5 Design Constraints 5](#_Toc319178990)

[3.6 Quality Characteristics 5](#_Toc319178991)

[4.0 Supporting Information 6](#_Toc319178992)

[4.1 Appendix A 6](#_Toc319178993)

4.2 Appendix B …………………………………………………………………………………………………………………………………………… 8

1. **Introduction**
   1. **Purpose**

The purpose of this document is to clarify and specify the capabilities of the grade book program known as "Dream Crusher". It will provide specific information regarding the functionality and typical use cases, and will be used by the developers as a guide to maintain a common end product goal. This document will establish the requirements needed for school faculty members, our stakeholders, who desire an application that facilitates the input of student rosters and grade book templates. This is an agreement between Doornail Technologies and our stakeholders, for the development of the grade book application.

* 1. **Scope**

The software application, "Dream Crusher", will allow the user to input student rosters and create grade book templates through the use of a graphical user interface. The high level system functionalities are described as follows:

* + 1. The system will import student rosters from a CSV file.
    2. The system will allow the creation and modification of grade book templates.
    3. The system will allow exporting of data to a spreadsheet format for later editing.
    4. The system will not be used to input grades.
  1. **Definitions, Acronyms, and Abbreviations**
     1. CSV Comma-separated values. This is a file format used to store tabular data in plain-text form. Entries are separated by commas. Typically, a CSV file is translated into a spreadsheet of rows and columns and is used in a program like Excel.
     2. Excel Microsoft Office Excel Program.
     3. GUI Graphical User Interface. This is the interface between the user and the software that will allow the user to tell the program what to do and to see the results displayed to the monitor.
     4. OS Operating System.
     5. SRS System Requirements Specification. This is used in reference to this specific document as a whole.

1. **General Description**
   1. **Product Perspective**

Our product is independent and totally self-contained. It is not a component of a larger product, and no databases will be accessed or needed for our product to function. The program manages its own functions of reading and saving CSV files, allowing the user to enter data manually, and exporting the data via CSV files for use in Excel.

CSV roster

Dream Crusher

CSV roster containing additional columns and formulas (for use in Excel)

* 1. **Product Function Summary**

The purpose of our product is to prepare a simple grade book layout for use in Excel.

* + 1. Dream Crusher will allow the user to import a CSV file containing a list of student names.
    2. The user will be able to modify the grade book templates which will be merged with the existing data from the roster CSV file.
    3. The program will save any merged data as an additional CSV file. The user can then open this CSV file in Excel, and the data will be spread into the appropriate columns.
  1. **User Characteristics**

The grade book user will require only minimal computer literacy skills with the Microsoft Windows line of operating systems. This includes mouse and keyboard usage, starting and exiting a program, and selecting files from a directory. Knowledge of how to use Microsoft Excel spreadsheets is also expected. The program will also provide specific details on how to use the grade book application via the help menu. The typical user will be an instructor, grader, or other person who uses Microsoft Excel to maintain their gradebook.

* 1. **General Constraints**
     1. The grade book application shall execute on all standard PC platforms running Microsoft Windows versions XP through Windows 7.
     2. The program shall not allow multiple instances of itself to run. This is done in order to ensure that no conflicts exist when exporting and saving data from the application.
     3. The program shall be written in the C# 4.0 NET framework, produced by Microsoft.
     4. The program shall be distributed as an install file, which can easily be run by double-clicking the file.
  2. **Assumptions and Dependencies**
     1. The professor is the only person who administers the grade book application; there is no student involvement.
     2. Data encryption and decryption techniques are not required since a password protected grade book system is not needed.

1. **Specific Requirements**
   1. **Functional Requirements**
      1. The program will take a CSV file as input, which contains the student roster. The student names will then be displayed within the application in a single column. These student names will be combined with the grade book template. If no template is selected, a default grade book template will be available for use.
      2. The program will allow the user the ability to merge one grade book template with one student roster file. The user can save the merged data at any time into a CSV file and store it in a specified directory. When the user saves the data to a CSV file, the program will prompt the user to name the CSV file.
      3. The program will allow the user to create, edit, and remove columns for assignments and exams within the grade book template. This layout will only contain columns for assignments and formulas.
      4. The program will allow the user to create and modify formula columns, each containing an arithmetic formula used to calculate the student overall scores and/or calculate weighted assignments in Excel.
      5. The program will take the specified formula and copy it into each cell of the formula column. The precondition is that one grade book template has already been merged with a student roster file.
      6. The formulas will not be imported as a list into the application (i.e. via a CSV file). They must be entered manually in the program when creating a new formula column.
      7. The program will not allow the user to enter the student grades for each assignment or exam. The scores will be entered through Excel, not the program (i.e. Dream Crusher).
      8. The application must display the name of the CSV file currently in use.
   2. **External Interface Requirements**
      1. The program will contain a GUI that will be the mediator between the user and the program. It will allow the user to give commands to the program and will display the contents of the grade book to the monitor.
      2. The GUI will contain all of the options which allow the user to manipulate or interact with the grade book application. These options will be accessible via a menu strip.
      3. The GUI will contain an option to exit the application.
   3. **Use Case Descriptions**
      1. Import the CSV-formatted student roster into the grade book application.
      2. Import an existing CSV-formatted grade book layout into the application.
      3. Merge Student Roster File With Grade Book Template.
      4. Create a new class assignment.
      5. Create a new formula column.
      6. Save the data to a new CSV file.
      7. Modify an existing formula column.

Refer to Appendix A (4.1) for more information.

Refer to Appendix B (4.2) for a preliminary user manual.

* 1. **Performance Requirements**
     1. The grade book application must load within 10 seconds after the initial startup.
     2. The initial importing of the CSV file containing the list of student names must finish within 10 seconds per 100 students in the file.
     3. Saving the data to a new CSV file must finish within 10 seconds per 100 students in the file.
  2. **Design Constraints**

The design of the grade book program must conform to the following constraints:

* + 1. All external files for use with the application will be stored as a CSV file.
    2. The design will be as object-oriented, in order to strive for easier maintainability.
  1. **Quality Characteristics**

The grade book will be designed and implemented according to the following quality standards:

* + 1. Formatting - The program will ensure that the CSV files are correctly formatted, and can be imported into Microsoft Excel without error. This is critical because the user should not be forced into personally scanning for formatting errors.
    2. Reliability - The program will be designed to be robust, minimizing any actions or events which could cause any crashes or loss of data.
    3. The program will adhere to the performance requirements mentioned in the performance section of this document.

1. **Supporting Information**
   1. **Appendix A: Use-Case Descriptions**
      1. Function: Import Student Roster
2. Actors: User
3. Description: The user will import the CSV-formatted student roster into the grade book application.
4. Pre-Conditions: The user has a CSV file containing a roster of student names, and has launched the program.
5. Post-Conditions: The user will have imported the student roster into the program.
6. Import Flow:
   1. User selects an option in the File menu to open a CSV roster.
   2. A directory window will pop up, prompting the user to select the CSV file to import.
   3. The user selects ‘Open’ and the program will import the student names to the form accordingly.
      1. Function: Import Existing Grade Book Layout
7. Actors: User
8. Description: The user will import an existing CSV-formatted grade book layout into the application.
9. Pre-Conditions: The user has an existing CSV file which represents the grade book layout consisting of assignment and formula columns.
10. Post-Conditions: The user will have imported an existing grade book template into the program.
11. Import Flow:
    1. User selects an option in the File menu to open a CSV grade book template.
    2. A directory window will pop up, prompting the user to select the CSV file to import.
    3. The user selects ‘Open’ and the program will import the grade book layout.
       1. Function: Merge Student Roster File With Grade Book Template
12. Actors: User
13. Description: Merge the roster file with the user template.
14. Precondition: User has imported a roster CSV file and grade book template CSV file into the application.
15. Post-conditions: The content of these two files are merged together into a separate CSV file.
16. Merge Flow:
17. The user selects ‘Merge’ option in the main menu.
18. A directory window will pop up, prompting user for the location and name of the merged file.
19. The user browses to the desired location and types the name for the file and clicks ‘OK’.
20. The file is merged and saved in the specified location.
    * 1. Function: Create a New Class Assignment
21. Actors: User
22. Description: The user will create a new class assignment.
23. Pre-Conditions: The user has an open grade book.
24. Post-Conditions: The user will create an assignment for the assignment column.
25. Create New Assignment Flow:
    1. User selects option to ‘Create New Column’.
    2. The user is prompted with a ‘Create New Assignment Column’ wizard.
    3. User completes the wizard with the appropriate assignment information.
    4. User selects ‘Finish’ in the wizard, and the form is populated accordingly.
       1. Function: Create New Formula Column
26. Actors: User
27. Description: Create one or more formula columns to represent grade scores.
28. Pre-Conditions: User must have an open grade book and one or more assignment columns.
29. Post-Conditions: The user will create a formula for the formula column.
30. Create New Formula Flow:
    1. User selects option to ‘Create New Column’.
    2. The user will be prompted with the ‘Create New Column’ wizard.
    3. The user will select the option ‘Create Formula Column’.
    4. User completes the wizard with the appropriate information, including the formula to be used.
    5. User selects ‘Finish’ and the form is updated accordingly. The formula from the formula column will be copied down to each cell in that column.
       1. Function: Saving Grade Book Template to a CSV File
31. Actors: User
32. Description: Save all the current template data as a CSV file.
33. Pre-Conditions: User must have an open grade book with one or more assignments and/or formula columns.
34. Post-Conditions: The Grade Book Template File is saved.
35. Saving Data to a CSV File Flow:
    1. User selects option to ‘Save’.
    2. A ‘Save As’ window will appear.
    3. User is prompted for the name and location of the file to be saved.
    4. User selects ‘Save’, and the file is saved to the chosen directory.
       1. Function: Editing an Existing Column
36. Actors: User
37. Description: Editing the formula/assignment on an already existing column.
38. Pre-Conditions: User must have an open grade book with one or more assignment or formula columns.
39. Post-Conditions: The column will be updated and displayed.
40. Editing an Existing Column Flow:
    1. User selects option to ‘Modify Column’.
    2. A ‘Modify Column’ wizard will appear.
    3. The wizard will prompt users to identify the assignment or formula column to be changed.
    4. The wizard will prompt for the new formula/assignment to be entered.
    5. User selects ‘Update’.
    6. New formula/assignment is displayed in the form accordingly.
    7. **Appendix B: User Guide**
       1. Opening the Program

To open the Dream Crusher Grade Book System, simply click on the executable file. The program will be ready for use within moments.

* + 1. Importing a Student Roster File

To import a student roster CSV file into the application, navigate to ‘File’, ‘Import’, ‘Roster File’. Select the CSV file which you would like to import, and select ‘OK’. The program will import the student names from roster and display them in the first column of the table.

* + 1. Importing a Grade Book Template

To import an existing grade book template from a CSV file, first select ‘File’, ‘Import’, then ‘Template’. Select the desired template from the window and click ‘OK’. The program will display the template information, such as assignment and formula columns in the table. There will be a default template associated with the program.

* + 1. Merging Roster and Template CSV Files

Note: Before merging the two files, you must have an existing roster file and an existing template file imported into the program.

To merge the roster and template CSV files, first select ‘Merge’ and then specify the location and the name of the new file where the contents should be merged to. The program will automatically merge the two files together and copy any formulas into their respective formula column cells.

* + 1. Creating a New Assignment Column

Note: In order to create a new assignment column, you must have an existing grade book template opened.

Select the option to ‘Create New Column’. A wizard will appear which will provide an additional option to ‘Create New Assignment’. Select this. Provide the name of the assignment and select ‘Finish’ once satisfied with your changes.

* + 1. Saving the Grade Book Template

To save the grade book template file, first select ‘File’, then ‘Save’. In the ‘Save As’ window that opens, navigate to the location where you wish to save your file and enter the desired file name, then click ‘Save’.

* + 1. Creating a New Formula Column

Note: In order to create a new formula column, you must have an open grade book with one or more assignments and/or formula columns.

Select the option to ‘Create New Column’. A wizard will appear which will provide an additional option to ‘Create New Formula’. Select this. Provide an Excel-style arithmetic formula, and select ‘Finish’ once satisfied with your changes.

* + 1. Editing an Existing Column

Note: In order to edit any column, you must have an open grade book with one or more assignment and/or formula columns.

Select the option to ‘Modify Column’. A wizard will appear which will prompt you to identify the formula/assignment column to be changed, and allow you to modify the existing content. Once you are satisfied with your changes, select the option to ‘Update’. The column will be updated and displayed.

* + 1. Exiting the Program

To exit the program, you can either click the ‘Exit’ icon located in the upper right corner of the program window, or you can select ‘File’, then ‘Exit’.  If you have unsaved changes, the program will prompt you with a window asking you if you’d like to save these changes.  If you select ‘No’, the program will exit immediately.  If you select ‘Yes’, the program will then give you the opportunity to navigate to the location you wish to save your file, enter the desired file name, then click ‘Save’.