Doornail Technologies, Inc.

System Requirement Specifications for the Grade Book System

Aaron Danielson | Luis Hidalgo | Timothy Jutras | Graham Williamson

CS3300 Software Engineering

12.03.2012

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Release Date | Responsible Party | Major Changes |
| 1.0 | 12.03.2012 |  | Release for design |

Table of Contents

[1.0 Introduction 3](#_Toc318805097)

[1.1 Purpose 3](#_Toc318805098)

[1.2 Scope 3](#_Toc318805099)

[1.3 Definitions, Acronyms, and Abbreviations 3](#_Toc318805100)

[1.4 References 3](#_Toc318805101)

[2.0 General Description 3](#_Toc318805102)

[2.1 Product Perspective 3](#_Toc318805103)

[2.2 Product Function Summary 3](#_Toc318805104)

[2.3 User Characteristics 3](#_Toc318805105)

[2.4 General Constraints 3](#_Toc318805106)

[2.5 Assumptions and Dependencies 4](#_Toc318805107)

[3.0 Specific Requirements 4](#_Toc318805108)

[3.1 Functional Requirements 4](#_Toc318805109)

[3.2 External Interface Requirements 4](#_Toc318805110)

[3.3 Use-Case Descriptions 4](#_Toc318805111)

[3.4 Performance Requirements 4](#_Toc318805112)

[3.5 Design Constraints 4](#_Toc318805113)

[3.6 Quality Characteristics 4](#_Toc318805114)

[3.7 Other Requirements 4](#_Toc318805115)

[4.0 Supporting Information 4](#_Toc318805116)

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.

* 1. Scope

The program to be developed is called Dream Crusher. The program will offer the user a graphical user interface to perform the following system functionalities:

* + 1. Manage one or more grade books for multiple classes, which may contain a variable number of students and assignments.
    2. A grade book must be able to support zero or more assignments; the user may add class assignments at any time.
    3. Submit individual student scores for all class assignments.
    4. Import and export excel data into and out of the grade book application.

A grade book allows an instructor to manage the individual student scores for class assignments, for which all the students are expected to complete.

* 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 more human-friendly rows and columns in a spreadsheet and used in a program like Excel.
     2. **SRS System Requirements Specification**. Used in reference to this document as a whole.
  2. References

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.

* 1. Product Function Summary
  2. 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. The program will also provide specific details on how to use the grade book application via the help menu.

* 1. General Constraints
     1. The grade book application shall execute on all standard PC platforms running Microsoft Windows versions XP through Windows 8.
     2. The program shall not allow multiple instances of itself to run. This is done 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.
      2. The program will allow the user to create columns for assignments and exams, which will contain the individual student scores.
      3. The assignments will not be imported as a list into the application, which title can be edited.
      4. The program will allow the user to create columns for arithmetic formulas to calculate the student overall scores.
   2. External Interface Requirements
      1. The program will be represented as a Windows form.
      2. All data will be shown in tabular format.
      3. All program actions will be accessible via the menu strip at the top of the form.
   3. Use-Case Descriptions
      1. Upload the CSV-formatted student roster into the grade book application.
      2. Create a new class assignment.
      3. Insert individual student scores on class assignments.
      4. Export data from the grade book application into an excel sheet.
      5. Import excel data into the grade book application.
   4. Performance Requirements
      1. The grade book application must load within 10 seconds after the initial startup.
      2. The excel data importing process must finish within 10 seconds.
      3. The excel data exporting process must finished within 10 seconds.
   5. Design Constraints
   6. Quality Characteristics
   7. Other Requirements

No other requirements have been identified.

1. Supporting Information

Section I: System Requirements

1. General
2. The program must run on Windows XP, Vista, and 7 Operating Systems.
3. The program must import and export all data in CSV format for use with Microsoft Excel.
4. The program must be able to support multiple grade books, each for a specific course.
5. The program must use a local configuration file for preferences, and the program must allow the user to specify the location of saved CSV files.
6. The program must be able to email students regarding new assignments and individual student scores.
7. For privacy reasons, the program must give the user the option to select which columns to view or export, i.e. omit student ID or other notes about the students.

Section II: Functional Requirements

1. Grade Book Functional Requirements
2. A grade book must have the instructor’s name as a required field.
3. A grade book must have a course ID as a required field, i.e. "CS 3300".
4. A grade book must be able to support zero or more assignments.
5. A grade book must allow the ability to add/remove a student to/from the student list.
6. In case a student has the same name as another student, a grade book must also contain a column of each student’s ID that corresponds to the student in order to differentiate between the same names.
7. Assignment Functional Requirements
8. An assignment must have an ID, title, and points available as required fields.
9. An assignment may have a description, type, and due date as optional fields.
10. An assignment must be editable, i.e. the title or point value can be changed manually.
11. Student Functional Requirements
12. A student must have an ID, email, first and last name as required fields.
13. A student may have a degree (undergraduate or graduate) and notes about the student as optional fields.
14. A student’s score will be maintained in a table of key and value pairs (assignment ID’s and points earned).
15. A student’s overall score will be represented by points earned divided by points available.

Section III: External Interface Requirements

1. Grade Book External Interface Requirements
2. A grade book must contain a list of all the students enrolled in the course in a single column.
3. A grade book must contain a column for each assignment and exam
4. A grade book must contain columns for the user created formulas, which can be used to calculate the intermediate and final grades.
5. A grade book must allow the ability to add/remove an assignment column or formula column.
6. The user will be able to omit the student ID column from view.
7. GUI (Graphical User Interface) External Interface Requirements
8. When the grade book application is opened, the GUI must be blank. In other words, no grade books or CSV files will be displayed.
9. The GUI must contain a File tab for options such as:
   1. Creating a new grade book.
   2. Saving a grade book layout.
   3. Opening a previously created grade book; a side-menu will contain the list of previously created grade books which are saved within a specific directory.
   4. Deleting a previously created grade book; a side-menu will contain the list of previously created grade books.
10. The GUI must contain a View tab for allowing the user to select which columns of the grade book to view.
11. The GUI must contain the traditional Close tab in the upper right hand corner, where upon being clicked will close the application.
12. The GUI must contain a tab for opening CSV files in order to choose which name list CSV file to import into the grade book.
13. A grade book layout and a CSV file must both be opened in the GUI in order for the GENERATE button to work.
14. Only one CSV file can be opened at a time in the GUI.
15. Only one grade book layout can be opened at a time in the GUI.
16. A Wizard will be included for grade book creation, which will, step-by-step, lead the user through all the data entry points.

Section IV: Performance Requirements

1. Grade Book Performance Requirements
2. The grade book table data must load within 10 seconds.
3. When the application is first executed, it must load within 10 seconds.
4. Excel data importing must finish within 10 seconds.
5. Excel data exporting must finish within 10 seconds.