# Gradebook Technical Documentation

Abby O'Neill, Maura O'Donnell, Timothy Moran

Gradebook Java application is composed of the following files:

openGradebook.java
loginPage.java
mainPage.java
createPage.java
rubricBuilder.java
pastRubric.java
Assignments class
GradebookGUI.java

---

**openGradebook.java**

- Contains main function and initiates login page

**loginPage.java**

- Button listener
  - Login
    - Compares user entered username and password from the text fields to saved username and password. If the username and password match what is saved, the login screen disappears and the user is taken to the main page. If the username and password entered do not match what is saved, the user is told that their input is incorrect and is prompted to try again.
- GUI Layout
  - The user is presented with empty text fields labelled for entering the username and password. There is a button for the user to click to login once the username and password have been entered.

**mainPage.java**

- Button listeners
  - createSemester
    - When the create semester button is pressed, the user is presented with a dialog box asking them to enter the semester name and year. Their answer is stored into a string with the spaces removed. A new directory is then created within the pre-existing semesters directory (stores all of the semesters) for this new semester. Within the new semester's directory, a file is created to store the list of classes for that semester. In the pre-existing semesters directory, if the semester list file does not yet

exist, it is created. The newly created semester is then appended to the semester list file.
    - createClass
        - When the create class button is pressed, the system checks if a semester has been created yet. If no semesters have been created, the user is prompted to create a semester before creating a class. If a semester has been selected, the user is brought to the create page.
    - Class buttons
        - A button is created for each class in the class list file of the selected semester. When a button is clicked for a class, the file path variables for the gradebook, rubric, and assignment files are set (which are then used in GradebookGUI.java). The user is then brought to the gradebook page where the saved information for that class in displayed.
- Important functions:
    - getSemesterList
        - Uses the Scanner class to read in the semester list file. Stores each line of the file (each semester) as an element in a List which is then converted to a string array which is returned. If the semester list file has not yet been created (no semesters created yet), the string array is set to “No semesters created”.
    - loadSemesters
        - Creates a combo box populated with the array returned from getSemesterList
    - getSelectedSemester
        - Takes the selected semester from the semester combo box as a parameter and returns the file path of that semester’s class list text file
    - getClassList
        - Reads the list of classes for the selected semester from that semester’s class list file and stores the class names into an array of strings which is then returned by the function
    - loadButtons
        - Using the array returned from getClassList, the function creates a button for each element of the array and adds a button listener with the command equal to the name of the class. Each class button is then added to the class list panel which is then set to be visible on the main page.
- GUI Layout
    - The main landing page brings the user to a page where they are given the option to create a new semester and to create a new class. If semesters have already been created, they will be displayed in a dropdown box next to the button to create a new semester. When a semester is selected from the drop down box, buttons with the name of each class in that semester are loaded in and displayed in the center of the main page. When the user clicks a button for

an already created class, they are brought directly to the gradebook page with that classes saved grades loaded in.

**createPage.java**

- Button listener
    - newRubric
        - Sets the file path variables for the gradebook, rubric, and assignment files using the current semester and course name and number. Appends the class list file of the current semester using BufferedWriter to add new course name and number to list. Brings user to rubricBuilder page.
    - pastRubric
        - Sets file path variables and appends course to class list file of the semester. Brings user to past rubric page.
    - Home
        - Returns the user to the main page
- Important functions:
    - getRubricList
        - Reads in list of saved rubrics using the scanner class and returns an array of strings which holds the semester and class name of the rubric. If the rubric file has not yet been created, the string array is set to “No rubrics created”. This function is used to populate the dropdown of past rubrics for the user to work from.
- Error checking functions:
    - emptyFields
        - Verifies that the course number and course name fields are filled
- GUI Layout
    - When creating a new class, the user is brought to a page with two empty text fields, one for the course number and one for the course name. The user is also presented with two buttons, one for creating a new rubric and and one for working from a past rubric which can be selected from the drop down menu next to the button.

**rubricBuilder.java**

- Contains the class “rubricBuilder”
- Important Functions:
    - getNumOfFields
        - Gets an integer from the user that determines how many rows of text boxes to have
    - ButtonClickListener
        - The function that does the error checking, writes the files, and eventually closes the window
- Error Checking Functions:
    - verifyPercentages

        - Verifies that the percentages entered by the user add up to 100
    - emptyFields
        - Verifies that there are no empty fields
- File I/O Functions:
    - writeRubricFile
        - Uses the FileWriter class and the BufferedWriter class to save the data entered by the user in the format of “assignmentType,percentage” (example: “Homework,50”)
    - writeAssignmentFile
        - Uses the FileWriter class and the BufferedWriter class to save the data entered by the user in a way that can be easily read later on to create all the columns for the gradebook page. The file ends up being the words ID, Last Name, and First Name (each on their own line) and then each assignment type (also each on their own line)
- GUI Layout:
    - The first thing you see is a message box that prompts you to enter the number of assignment types you want to use for the class. After the field number is determined, the main screen for rubric creation is shown. Here, the user enters the name of the assignment type, and the percentage weight for that assignment type. There are four panels here, a header panel (that obviously contains the header), the assignment panel, the percentage panel (both are y-axis box layout panels; looks like a grid of text boxes with two columns and number of rows matching the number previously entered. Once values are entered for each text box and the button is pressed, the error checking functions are called, and if there are no errors, the file I/O functions are called and you are brought to the gradebook page.

**pastRubric.java**

- Contains the class “pastRubric”
- Important Functions:
    - getRubric
        - Uses the scanner class to find the data for the past rubric specified by the user, and return a string array containing that data
    - splitRubricArray
        - This takes the array that getRubric returned and goes through it, saving the contents of each line to two new arrays, one for assignments and one for grades, that are later used to populate the grid with the data from the past rubric
    - ButtonClickListener
        - The function that does the error checking, writes the files, and eventually closes the window
- Error Checking Functions:
    - verifyPercentages

        - Verifies that the percentages entered by the user add up to 100
    - emptyFields
        - Verifies that there are no empty fields
- File I/O Functions:
    - writeRubricFile
        - Uses the FileWriter class and the BufferedWriter class to save the data entered by the user in the format of “assignmentType,percentage” (example: “Homework,50”)
    - writeAssignmentFile
        - Uses the FileWriter class and the BufferedWriter class to save the data entered by the user in a way that can be easily read later on to create all the columns for the gradebook page. The file ends up being the words ID, Last Name, and First Name (each on their own line) and then each assignment type (also each on their own line)
- GUI Layout
    - Exactly the same as the layout for creating a new rubric. The only differences are that there is no initial window asking for the number of assignment types, and all the text boxes are initially filled (but can still be edited by the user)

**Assignments class**

- Private data members assignment(string) and percentage(int)
- Constructor that initializes assignment and percentage
- Accessor methods for assignment and percentage
- Mutators methods for assignment and percentage

**GradebookGUI.java**

- GUI layout
    - Menubar (JMenuBar) at the top of the page with items(JMenuItem) under each tab (JMenu) the user can select if he/she wants to change data within the gradebook, calculate average grades (JRadioButtonMenuItem), or save the file
    - Datatable (JTable) that displays the data for the grade book (student name, student ID, and grades)
    - Title of the class (JLabel)
- File I/O Functions
    - initGradeBook: opens csv file and initiates gradebook cell values
    - writeGradebookFile: writes new file from data in the grade book datatable cell
    - saveAssignmentFile: writes new file from data in the grade book datatable cell with an additional assignment
    - fileRemoveAssignment: writes new file without a specific assignment
- Contains functions to change the grade book datatable
    - AddAssignment- Allows user to select a particular assignment from the rubric and add that assignment to the grade book

    - RemoveAssignment- Removes the user-selected column that is not their first name, last name, or ID number
    - AddStudent- Adds a row to the bottom of the program
    - RemoveStudent- Removes the user-selected row
    - CalculateStudentAverage- Adds an additional column and calculates the student's average based on the weights given in the rubric
    - CalculateAssignmentAverage- Adds an additional row and calculates the average for each assignment
- Other functions
    - getRubric: when called, open rubric file and returns the rubric information
    - setAssignments: creates an array of assignment objects and assigns each assignment a specific weight and name
    - getColumns: when called, opens assignment file and return an array of column names