# Project Analysis

**Group 2:**
John Abueg, William Rios Crespo, Joshua Kerley, Michael Lancaster
CMSC 495 63802

## Author Note:

Document Version PA008
Date: 2018-10-12

# Version Control

| Document | Date | Action | Name | Email |
|---|---|---|---|---|
| PA001 | 2018-09-02 | Created | John Abueg | j.abueg13@gmail.com |
| PA002 | 2018-09-04 | Modified | John Abueg | j.abueg13@gmail.com |
| PA003 | 2018-09-06 | Modified | William Rios Crespo | william.rioscrespo19@gmail.com |
| PA004 | 2018-09-08 | Modified | Josh Kerley | jkillakerlz@gmail.com |
| PA004.1 | 2018-09-08 | Revised | Josh Kerley | jkillakerlz@gmail.com |
| PA004.2 | 2018-09-08 | Revised | John Abueg | j.abueg13@gmail.com |
| PA005 | 2018-09-09 | Modified | John Abueg | j.abueg13@gmail.com |
| PA006 | 2018-09-09 | Modified | Josh Kerley | jkillakerlz@gmail.com |
| PA006.1 | 2018-09-09 | Revised | William Rios Crespo | william.rioscrespo19@gmail.com |
| PA006.2 | 2018-09-09 | Reviewed | John Abueg<br>Josh Kerley<br>Michael Lancaster<br>William Rios Crespo | j.abueg13@gmail.com<br>jkillakerlz@gmail.com<br>william.rioscrespo19@gmail.com<br>lancastermc@gmail.com |
| PA007 | 2018-09-13 | Modified | John Abueg | j.abueg13@gmail.com |
| PA007.1 | 2018-09-29 | Revised | John Abueg | j.abueg13@gmail.com |
| PA008 | 2018-10-12 | Modified | John Abueg | j.abueg13@gmail.com |

# Grade Calculator Project Analysis:

1. **Outside Systems:** The system shall interface with the following outside systems:
    a. User
    b. A .dat file on the system computer that can be written to and read from.

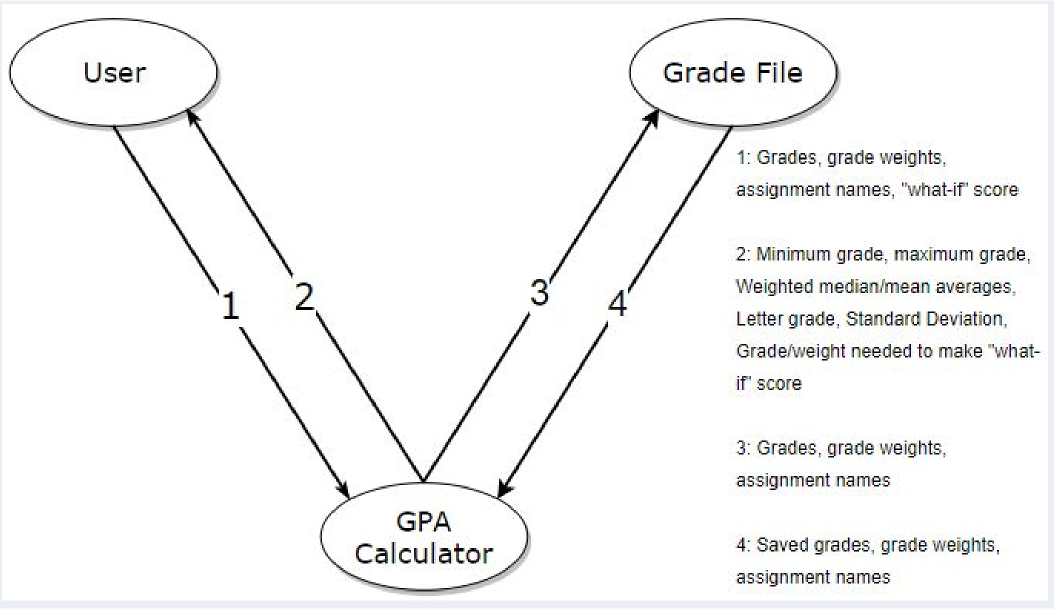
2. **Input Data and Source(s):** The system shall receive the following data:
    a. Assignment name
    b. Assignment weight
    c. Assignment grades
    d. Desired grade for “what-if” purposes (e.g. “What do I need to score to make to get an 85 overall for this class?”)

    All input data comes from either the user or a specified .dat file.

3. **Output data and Destination(s):** The system shall send the following data:
    a. All inputted grades for a student
    b. All assignment names if given
    c. All assignment weights
    d. The highest (“maximum”) of all current entered scores
    e. The lowest (“minimum”) of all current entered scores
    f. Mean average of current entered scores after adjusted for weight
    g. Median average of current entered scores after adjusted for weight
    h. Standard deviation of current entered scores after adjusted for weight
    i. A letter grade based on the mean average
    j. Average score and weight needed to meet desired “what-if” final grade

    All output data is displayed to the user on the monitor. Data in bullet points a-c above can also be saved to a specified .dat file. Sending to both output locations may be performed on the same data.

    See Figure 1: Context Diagram



1: Grades, grade weights, assignment names, "what-if" score

2: Minimum grade, maximum grade, Weighted median/mean averages, Letter grade, Standard Deviation, Grade/weight needed to make "what-if" score

3: Grades, grade weights, assignment names

4: Saved grades, grade weights, assignment names

*Figure 1- Context Diagram*

4. **Data processing:** The user inputs data manually OR loads a file:
    a. The following are calculated based on grade and weight (if applicable) input data.
        i. Minimum grade score
        ii. Maximum grade score
        iii. Weighted mean average: Each grade is multiplied by its weight for the grade and added together, then that total is divided by the number of weight units.
        iv. Weighted median average
            1. Each grade is given representation equal to its weight.
            2. The grades are placed in numerical order.
            3. If the number of data points are odd the following formula is used to find the median's place: ([the number of data points] + 1) ÷ 2
            4. If the number of data points are even, the data points in the following places are needed: (number of data points/2), (number of data points/2)+1
        v. Standard deviation:
            1. Find the Mean
            2. For each data point, subtract the Mean and square the result
            3. Find the Mean of those squared differences.
            4. Find the square root of that Mean.
        vi. Letter grade: Find the Mean and represent it as a letter grade based on the following scale: A (90-100), B (80-89), C (70-79), D (60-69), F (<60). Plus/minus signs will not be given with the letter grade.
        vii. "What-if" score and weight:

1. Follow the Mean Average process, but add a data point based on the target score.
   a. If target score is lower than mean, use: Grade of 0, Weight 1.
   b. If target score is higher than mean, use: Grade of 100, Weight 1.
2. Evaluate the result.
   a. If the result exceeds the target “what-if” score then REDO steps 1 and 2, except 1 is added (if data point grade is 0) or subtracted (if data point grade is 100) from the grade number with every loop.  Once a data point grade is found that does not meet the “what-if” grade, this loop stops and the last successful data point (i.e. current data point grade+1) is returned as the target score with a weight of 1.
   b. If the result does not meet or exceed the target “what-if” score:
      i. Add another data point of the type chosen in step 1.
      ii. Perform the Mean Averages process
      iii. Re-evaluate.
      iv. Repeat steps 2.b.i-2.b.iii until the result exceeds the target score.  **Keep track of the number of data points added to the original grades.**
      v. Once the target score is exceeded, repeat steps 2.b.ii-2.b.iii with 1 added/subtracted from the last data point’s grade with every loop until the mean no longer meets the “what-if” grade.
      vi. Once a data point grade is found that does not meet the “what-if” grade, this loop stops and all the 100 grades plus the current data point grade+1 are added together.
      vii. The total of the added grades are divided by the number of added data points to give us the average “what-if” score needed.  The number of added data points is the “what-if” weight.

b. Calculated results are displayed on screen.  This includes the “What-if” target score and weight if requested
c. Grades, grade names, and grade weight may be saved to file

5. **Subsystems Data:** The following have been identified as subsystems that shall facilitate data flow:
   a. Input: This subsystem allows the user to enter grade data to create an event using a combination of input text, drop down menus and buttons.  Loading grade data from a file is also handled here.

b. GUI: This subsystem handles the graphical information for both user input and display via various JButtons and JPanels.  This class also performs the calls to send/receive data to the other subsystems.
c. Calculator: This subsystem receives the values from the GUI subsystem and will perform the mathematical operations needed to perform minimum grade, maximum grade, averaging, letter grade, and standard deviation calculations.
d. Output: This subsystem can receive specific values from the GUI subsystem and write them to a file.

See Figure 2 - Subsystem Diagram

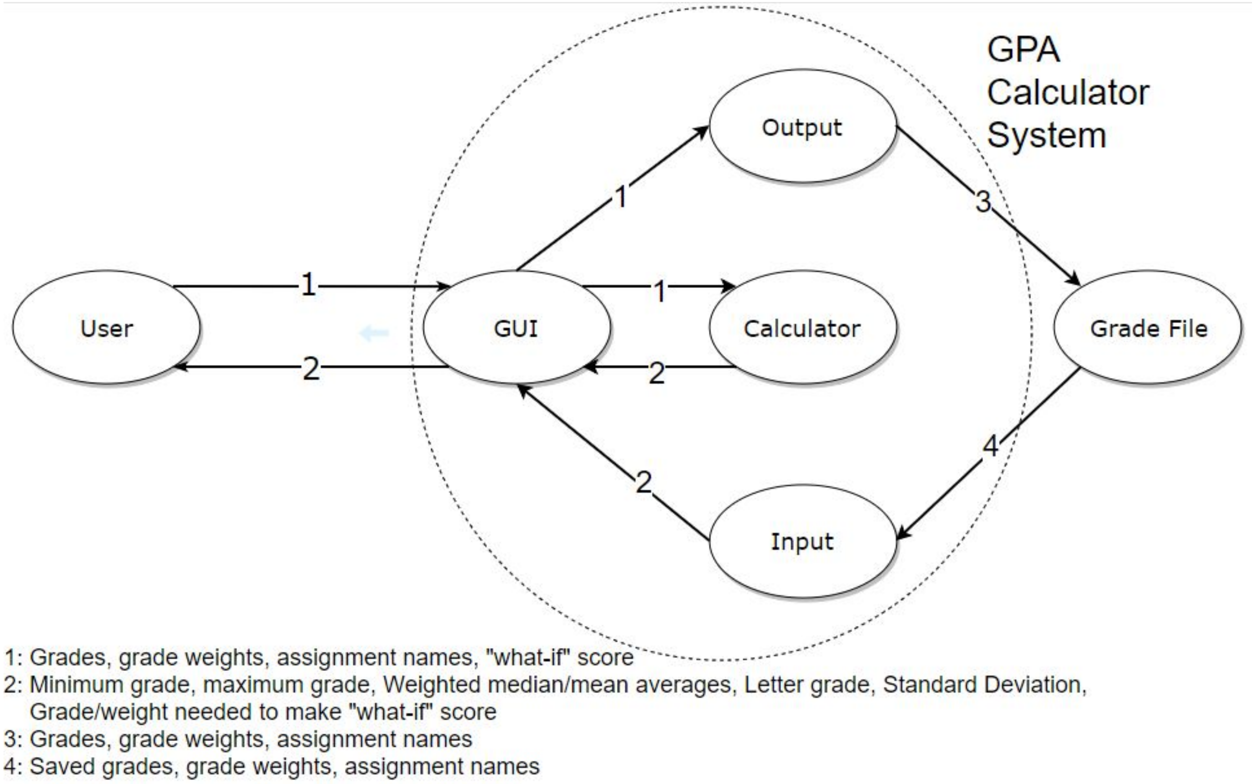

GPA Calculator System
Output
User
GUI
Calculator
Grade File
Input
1
1
1
2
2
2
3
4
1: Grades, grade weights, assignment names, "what-if" score
2: Minimum grade, maximum grade, Weighted median/mean averages, Letter grade, Standard Deviation, Grade/weight needed to make "what-if" score
3: Grades, grade weights, assignment names
4: Saved grades, grade weights, assignment names

*Figure 2 - Subsystem diagram*

## 6. Requirement implementation via subsystems:

| Requirement | Subsystem/Description |
|---|---|
| 1 | Input - shall accept assignment names, weight, and grade. |
| 2 | File - This system shall save accepted inputs. |
| 3 | File - This system shall load saved data. |
| 4 | Calculator - This system shall calculate grade based off of data. |

| | |
|---|---|
| 5 | Input - This system shall NOT accept letter grades. |
| 6 | Input - This system shall NOT accept invalid inputs such as special characters. |
| 7 | GUI, Calculator -This system shall display maximum, minimum, mean, median, and standard deviation scores. |
| 8 | File, Calculator - This system shall contain a ‘what if’ scenario option to calculate hypothetical grades. |
| 9 | GUI - This system shall provide a graphics user interface for the ease of use for the user. |

7. **Risk analysis:**

| Identify risks | Risk mitigation |
|---|---|
| A user selects an incorrect input file that doesn't resemble the data file required. | Limit file selection to only .dat files. If the .dat file selected doesn’t contain the required guidelines, an exception will be thrown by the application to notify the user to pick a valid record. |
| Invalid inputs such as special characters, letter grades, negative scores or grades higher than one hundred. | The application will throw and catch an exception that will open a dialog window that will notify that the user entered an invalid input. |
| Overwriting an incorrect file. | Limit file selection to only .dat files and display a dialog window to confirm overwriting the selected file. |

8. **Possible project enhancements:**
    a. Modify the grade calculator into a classroom grade book that contains a list of students that calculates the scores for each student.
    b. Query individual student results.
    c. Rank student results based on their aggregate grade.
    d. Export grades to an email address.
    e. Have a user notification that reports when a student GPA falls below a 2.0.
    f. Follow up messages that provide weekly reports to users.
    g. Alter the Java interface into an HTML file that could be accessed from a remote site while implementing a MySQL database to create, maintain and search the records.