

Combined test cases for review

Test Case 1.1

Purpose

Verify: *This test does not verify either of these things*

1. The product outputs a CSV file with the results of the cumulative grading
2. The product outputs the correct format CSV file

Requirement Traceability:

Requirements: 1.1, 1.2, 1.3, 1.4, 1.5 Setup:

1. Download *to where?* the latest version of Lambda Software's Exam Grading software from the wiki download page.
2. Make sure you have a copy of a test file with a key followed by student tests (.pdf), a PolyLearn roster file (.csv), and all oracle files which you will compare to *where do I get these?*
3. Find the location in your directory where the executable JAR file is located *where do I put them?*
4. Double click on the JAR file to run the application *OR java -jar ..*
5. Follow specific instruction from the Test Data section below
6. Compare the output files to the oracle files and make sure there are no differences

Test Data: *You MUST provide directions for using a tool to do this.*

Action	Input	Expected Output
Application <i>start</i> Bootstrap	java -jar	Application's GUI window pops up <i>How do I verify it's correct?</i>
Initiate PDF file selection	Click "Browse"	File Browser window pops up <i>in what directory?</i>
Select a non-PDF file	Select the TSV file raw.tsv <i>click Open</i>	Verify that an error indicates you have chosen a file of the wrong format. <i>just give expected output be specific</i>
Choose PolyLearn CSV file	Click "Browse"	File Browser window pops up
Select CSV file	Click on provided PolyLearn file <i>be specific</i>	File Browser pop up closes and <u>file name</u> shows in field <i>which name, exactly</i>
Move to next page	Click "Set Output Parameters"	The next page of the GUI will appear <i>How do I know it's correct?</i>
Begin scanning tests	Click "Begin Scan"	Informs you that your files have finished being scanned <i>Exactly what should it say?</i>

Test Case 1.4

Purpose

Verify:

1. Separate PDF files input at the same time will be treated as different versions of the same test.

Requirement Traceability:

Requirements: 1.4 Setup:

1. Download the latest version of Lambda Software's Exam Grading software from the wiki download page.

2. Make sure you have a copy of a test file with a key followed by student tests (.pdf), a PolyLearn roster file (.csv), and all oracle files which you will compare to
3. Find the location in your directory where the executable JAR file is located
4. Double click on the JAR file to run the application
5. Follow specific instruction from the *Test Data* section below
6. Compare the output files to the oracle files and make sure there are no differences

Test Data:

Same problems as above

Action	Input	Expected Output
Application Bootup		Application's GUI window pops up
Initiate PDF file selection	Click "Browse"	File Browser window pops up
Select a PDF file	Select the version1.pdf and version2.pdf	File Browser pop up closes. Verify both file names appear in the PDF field.

Test Case 1.5

Purpose

Verify:

1. The Cal Poly PolyLearn roster CSV file is required.

Requirement Traceability:

Requirements: 1.5 Setup:

1. Download the latest version of Lambda Software's Exam Grading software from the wiki download page.
2. Make sure you have a copy of a test file with a key followed by student tests (.pdf), a PolyLearn roster file (.csv), and all oracle files which you will compare to
3. Find the location in your directory where the executable JAR file is located
4. Double click on the JAR file to run the application
5. Follow specific instruction from the *Test Data* section below
6. Compare the output files to the oracle files and make sure there are no differences

Test Data:

Action	Input	Expected Output
Application Bootup		Application's GUI window pops up
Initiate PDF file selection	Click "Browse"	File Browser window pops up
Select a PDF file	Select the PDF file version1.pdf	File Browser pop up closes and file name shows in field. Verify that the Next button cannot be clicked until a roster file is selected.
Choose PolyLearn CSV file	Click "Browse"	File Browser window pops up
Select CSV file	Click on provided PolyLearn file	File Browser pop up closes and file name shows in field. Verify that the Next button can now be clicked. <i>How?</i>

Test Case 2

be specific

what should happen?

*how?
which?*

Purpose

No, it doesn't

Verify:

- 1. The product outputs a CSV file with the results of the cumulative grading
- 2. The product outputs the correct format CSV file

Requirement Traceability:

Requirements: 2, 2.1

Setup:

- 1. Download the latest version of Lambda Software's Exam Grading software from the wiki download page.
- 2. Make sure you have a copy of a test file with a key followed by student tests (.pdf), a PolyLearn roster file (.csv), and all oracle files which you will compare to
- 3. Find the location in your directory where the executable JAR file is located
- 4. Double click on the JAR file to run the application
- 5. Follow specific instruction from the *Test Data* section below
- 6. Compare the output files to the oracle files and make sure there are no differences

Test Data:

Action	Input	Expected Output
Application Bootup		Application's GUI window pops up
Choose testing PDF file	Click "Browse"	File Browser window pops up
Select PDF file	Click on provided test file	File Browser pop up closes and file name shows in field
Choose PolyLearn CSV file	Click "Browse"	File Browser window pops up
Select CSV file	Click on provided PolyLearn file <i>click OK</i>	File Browser pop up closes and file name shows in field <i>what?</i>
Move to next page	Click "Set Output Parameters"	The next page of the GUI will appear <i>what does it display?</i>
Begin scanning tests	Click "Begin Scan"	Informs you that your files have <u>finished</u> being scanned

How can they be finished when I just clicked "Begin..."? Seems like a step is missing.

Test Case 3

Purpose

Verify:

- 1. The product outputs a CSV file for each test version
- 2. The CSV files have the correct format as specified in the SRS:

Requirement Traceability:

Requirements: 3,3.1

Setup:

- 1. Download the latest version of Lambda Software's Exam Grading software from the wiki download page.
- 2. Make sure you have two pdfs of two versions of a test, a PolyLearn roster file (.csv), and all oracle files

- which you will compare to
- Execute the jar file per the instructions below
 - Compare the output files to the oracle files and make sure there are no differences

Test Data:

You didn't describe an output

Action	Input	Expected Output
run the jar file from command line	java -jar TestGrading -i test1v1.pdf,test1v2.pdf -r roster.csv	test1v1.csv and test2v2.csv should have the correct version results for the tests and exist in the same folder as the pdfs

How will the tester determine this?

Test Case 4.1

Purpose Verify that files are stored in the working directory if no save directory is specified.

Requirement Traceability: Requirement 4.1

Also, using a base name

Setup:

- Download the latest version of Lambda Software's Exam Grading software from the wiki download page.
- Make sure you have a copy of a test file with a key followed by student tests (.pdf) and a PolyLearn roster file (.csv)
- Find the location on your drive where the executable JAR file is located. Write down this location.
- Double click on the JAR file to run the application
- Follow specific instruction from the *Test Data* section below

No, you specify.

Test Data:

Action	Input	Expected Output
Select the pdf file	Click the button to select a pdf and browse to the location of the pdf on your computer	The pdf text box fills with the location of the pdf
Select the csv file	Click the button to select a csv and browse to the location of the csv on your computer	The csv text box fills with the location of the csv
Go to next input page	Click the select output parameters button	The output parameters window replaces the input parameters window <i>what does it say?</i>
Leave the output directory parameter blank	Nothing	Nothing
Choose a unique base filename	<i>No - don't make tester think.</i> Pick a basename using the date and time, such as 527442, write it down, and enter it into the basename text box	The text box fills with your <i>selected basename</i> — 527442
Start the scan	Click start scan	The program begins to scan and shows a progress window <i>what does it look like?</i>
Wait for the scan to finish	Nothing... maybe get a snack	The program eventually indicates that the scan is complete <i>how?</i>
Check the working directory for the output files	<i>Specify</i> Open up the location you wrote down as part of the setup. Look in this directory for a file named e.g. 527442_cumulative.csv.	If present, the test was successful.

Not an expected output.

Test Case 4.2.1

Purpose Verify that supplying no base filename will result in output files that use the base name of the input pdf as the base filename. *And supplying a directory for output*

Requirement Traceability: Requirement 4.2.1

Setup:

1. Download the latest version of Lambda Software's Exam Grading software from the wiki download page.
2. Make sure you have a copy of a test file with a key followed by student tests (.pdf) and a PolyLearn roster file (.csv)
3. Find the location on your drive where the executable JAR file is located.
4. Double click on the JAR file to run the application
5. Follow specific instruction from the *Test Data* section below

Test Data:

Action	Input	Expected Output
Select the pdf file	Click the button to select a pdf and browse to the location of the pdf on your computer	The pdf text box fills with the location of the pdf
Select the csv file	Click the button to select a csv and browse to the location of the csv on your computer	The csv text box fills with the location of the csv
Go to next input page	Click the select output parameters button	The output parameters window replaces the input parameters window
Select an output directory	Click the select output directory button and browse to a directory where your user has write permission. Write down this location. <i>Specify</i>	The output directory text field fills with the name of the output directory you chose
Do not select a base filename	Nothing	Nothing
Start the scan	Click start scan	The program begins to scan and shows a progress window
Wait for the scan to finish	Nothing... maybe get a snack <i>same problem as above</i>	The program eventually indicates that the scan is complete
Check the working directory for the output files	Open up the location you wrote down as part of the previous step. Look in this directory for a file named e.g. 527442_cumulative.csv.	Iff present, the test was successful.

Test Case 4.2.2 *How is this different from 4.1?*

Purpose: Verify that a basename input will be used in the storage of output files by the user.

Requirement Traceability: Requirement 4.2.2

Setup: 1. Download the latest version of Lambda Software's Exam Grading software from the wiki download page.

2. Make sure you have a copy of a test file with a key followed by student tests (.pdf) and a PolyLearn roster file (.csv)
3. Find the location on your drive where the executable JAR file is located.
4. Double click on the JAR file to run the application
5. Follow specific instruction from the *Test Data* section below

Test Data:

Same problems as above

Action	Input	Expected Output
Select the pdf file	Click the button to select a pdf and browse to the location of the pdf on your computer	The pdf text box fills with the location of the pdf
Select the csv file	Click the button to select a csv and browse to the location of the csv on your computer	The csv text box fills with the location of the csv
Go to next input page	Click the select output parameters button	The output parameters window replaces the input parameters window
Select an output directory	Click the select output directory button and browse to a directory where your user has write permission. Write down this location.	The output directory text field fills with the name of the output directory you chose
Choose a base file name	Type the desired name (gradedTests)	The output text field will be called gradedTests.pdf
Start the scan	Click start scan	The program begins to scan and shows a progress window
Wait for the scan to finish		The program indicates that the scan is complete
Check the working directory for the output files	Open up the location you wrote down as part of the previous step. Look in this directory for a file named gradedTests_cumulative.csv.	If present, the test was successful.

Test Case 4.3

Purpose Confirm that if files of the same name exist in the working directory, the user will be asked if they would like to delete the files.

Requirement Traceability: Requirement 4.3

Setup: 1. Download the latest version of Lambda Software's Exam Grading software from the wiki download page.

2. Make sure you have a copy of a test file with a key followed by student tests (.pdf) and a !PolyLearn roster file (.csv)
3. Create two files, one called gradedTests_cumulative.csv and gradedTests.csv. Save these in the same directory as the JAR file.
4. Find the location on your drive where the executable JAR file is located.
5. Double click on the JAR file to run the application
6. Follow specific instruction from the *Test Data* section below

Test Data:

Same problems as above

Action	Input	Expected Output
Select the pdf file	Click the button to select a pdf and browse to the location of the pdf on your computer	The pdf text box fills with the location of the pdf
Select the csv file	Click the button to select a csv and browse to the location of the csv on your computer	The csv text box fills with the location of the csv
Go to next input page	Click the select output parameters button	The output parameters window replaces the input parameters window

Select an output directory, the working directory	Click the select output directory button and browse to the working directory. Write down this location.	The output directory text field fills with the name of the output directory you chose
Select next	Click the next button	The program should output that there is an error, with the files being in the selected directory already

Test Case 5

Purpose:

Verify that the program provides outward indicator of progress in the form of a progress bar once test grading has been initiated.

Sounds different than Purpose I don't think this verifies what the Purpose describes

Requirement Traceability:

Functional requirement #5.

Setup:

1. Download the latest version of Lambda Software's Exam Grading software from the [wiki download page](#).
2. Make sure you have two files on hand:
 - a. A copy of an exam file with a key followed by multiple student exams (.pdf)
 - b. A student roster file (.csv)
3. Find the location in your directory where the executable JAR file is located
4. Double click on the JAR file to run the application
5. Follow specific instructions from the *Test Data* section below

If all actual output aligns with the information in the "Expected Output" column for each step, this test case is considered passing.

Test Data:

Same problems as above

Action	Input	Expected Output
Select .pdf exam file	Press the "Browse" button and browse your computer for the appropriate PDF file.	The PDF text box will fill with the path name of the selected file.
Select .csv roster file	Press the "Browse" button and browse your computer for the appropriate CSV file.	The CSV text box will fill with the path name of the selected file.
Go to the next step	Press the "Setup output parameters" button.	The page to select output parameters will replace the input selection page.
Start the grading	No output directory nor file prefix need to be selected; press the "Begin Grading" button.	The scan progress page will replace the output selection page. <i>what does it display?</i>
Verify progress bar updates	N/A	As the grading progresses, the progress bar on the final page should fill up relative to the percentage of student exams that have finished grading.
Verify progress bar finishes	N/A	Once the grading is finished, the progress bar should be full and the "Open results directory" and "Done" buttons should activate. <i>OK</i>

how do I verify this?