Team Lynx – Test Cases

Team peach specified 7 software requirements in their documentation for Peachy Galaxy:

* Loading data from file
* Error Processing
* Applying filters
* Using a custom sort order
* Visualizing data
* Printing to file
* Exporting to PDF

Team Lynx added the features below to the Peachy Galaxy:

* Saving application state
* Two new ways to visualize data (graphs)
* Column Sorting

Based on these requirements, we created test cases for various possible user test cases documented below. Test cases were developed in two methods: first, programmatically writing test cases using the QTest Library; second, manually using the GUI to replicate users actions. All programmatically written test cases can be found in the /test folder of the Peachy Galaxy project, and the ‘unit to test row’ in the following charts will provide the name of the method which was called in that test case.

Test cases that:

Testcase ID: 023, 021 (Both minor user experience issues)

|  |  |
| --- | --- |
| **Test Case ID:** | 001 |
| **Unit to test:** | QSORT\_LIST\_01 |
| **Requirements:** | 3.1.1 Final System Delivery (Loading data from file) |
| **Assumptions:** | QSortListIO will catch and throw exception when given an invalid location. |
| **Test Data:** | “NO FILE” |
| **Steps to be executed:** | Pass string to sorter, then call method .readList() and store  output as a QList |
| **Expected Result:** | QSortListIO success |
| **Actual Result:** | Debugger error |
| **Pass/Fail:** | Fail |
| **Comments:** | QSortListIO should contain a catch for invalid strings passed to the  class. |
| **Solution:** | Design a try/catch statments for invalid files locations. |

|  |  |
| --- | --- |
| **Test Case ID:** | 002 |
| **Unit to test:** | QSORT\_LIST\_02 |
| **Requirements:** | 3.1.1 Final System Delivery (Loading data from file) |
| **Assumptions:** | QSortListIO will catch and throw exception when given an invalid location. |
| **Test Data:** | NULL |
| **Steps to be executed:** | Pass string to sorter, then call method .readList() and store  output as a QList |
| **Expected Result:** | QSortListIO success |
| **Actual Result:** | Debugger error |
| **Pass/Fail:** | Fail |
| **Comments:** | QSortListIO should contain a catch for invalid strings passed to the  class. |
| **Solution:** | Design a try/catch statments for invalid files locations. |

|  |  |
| --- | --- |
| **Test Case ID:** | 003 |
| **Unit to test:** | QSORT\_LIST\_03 |
| **Requirements:** | 3.1.1 Final System Delivery (Loading data from file) |
| **Assumptions:** | QSortListIO will catch and throw exception when given an invalid location. |
| **Test Data:** | Grants\_expanded.csv |
| **Steps to be executed:** | Pass string to sorter, then call method .readList() and store  output as a QList |
| **Expected Result:** | QSortListIO success |
| **Actual Result:** | Success |
| **Pass/Fail:** | Pass |
| **Comments:** | QSortListIO should contain a catch for invalid strings passed to the  class. |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 004 |
| **Unit to test:** | QSORT\_LIST\_04 |
| **Requirements:** | 3.1.1 Final System Delivery (Loading data from file) |
| **Assumptions:** | QSortListIO will catch and throw exception when given an invalid location. |
| **Test Data:** | test\_file.txt |
| **Steps to be executed:** | Create text file, Pass string to sorter, then call method  .readList() and store output as a QList |
| **Expected Result:** | QSortListIO success |
| **Actual Result:** | Success |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 005 |
| **Unit to test:** | QCSV\_READER\_TEST\_09 |
| **Requirements:** | 3.1.1 Final System Delivery (Loading data from file) |
| **Assumptions:** | CSVReader will catch and throw exception when given an invalid  format |
| **Test Data:** | STAR 2015 Acuity STAR Western CS intro V2.pptx |
| **Steps to be executed:** | Pass string of location to CSVReader, then use method.  getData and pass it to vector, then check size to validate that it was accepted |
| **Expected Result:** | CVSReader catch invalid file |
| **Actual Result:** | Failed. CVSReader did not check file format |
| **Pass/Fail:** | Fail |
| **Comments:** | CSVReader should not be reliant upon the user the enter a .csv file.  There should be a try catch so the program does not crash |
| **Solution:** | Design a simple check to validate CSV files. |

|  |  |
| --- | --- |
| **Test Case ID:** | 006 |
| **Unit to test:** | Loading incorrect file from GUI |
| **Requirements:** | 3.1.1 Final System Delivery (Loading data from file) |
| **Assumptions:** | Error thrown on non CVS |
| **Test Data:** | Teaching\_test.docx |
| **Steps to be executed:** | 1. Select a tab 2. Click ‘load file’ 3. Pick an invalid CVS type 4. Wait for program response |
| **Expected Result:** | Program will not load invalid data |
| **Actual Result:** | Program did not load invalid file. |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 007 |
| **Unit to test:** | Loading incorrect tab types |
| **Requirements:** | 3.1.1 Final System Delivery (Loading data from file) |
| **Assumptions:** | Error thrown on wrong CSV tab type |
| **Test Data:** | Grants\_expanded.csv, GrantsClinicalFunding\_sample.csv,  Presentation\_expanded.csv, Presentation\_sample.csv, Program\_Teaching\_expanded.csv,  Publication\_expanded.csv, Publications\_sample.csv, Teaching\_sample.csv |
| **Steps to be executed:** | 1. Select a tab 2. Click ‘load file’ 3. Pick an invalid CVS type 4. Wait for program response |
| **Expected Result:** | Program will not load invalid tab type data |
| **Actual Result:** | Program did not load incorrect CVS file type. |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 008 |
| **Unit to test:** | Clicking Edit Button |
| **Requirements:** | 3.1.2 Final System Delivery (Error Proccessing) |
| **Assumptions:** | When clicking the Edit Button, Peachy galaxy should prompt me with the EditFields dialog. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Click edit button |
| **Expected Result:** | EditField dialog box will open |
| **Actual Result:** | EditField dialog box will open |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 009 |
| **Unit to test:** | Clicking Export Button |
| **Requirements:** | * 3.1.8 Final System Delivery (Exporting to PDF) |
| **Assumptions:** | When clicking the Export Button Peachy galaxy should prompt me with the “Export File” dialog. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Click export button |
| **Expected Result:** | “Export File” dialog box will open |
| **Actual Result:** | “Export File” dialog box will open |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 010 |
| **Unit to test:** | Clicking Print Button |
| **Requirements:** | 3.1.7 Final System Delivery (Printing to file) |
| **Assumptions:** | When clicking the Print Button Peachy galaxy should prompt me with the “Print” dialog. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Click print button |
| **Expected Result:** | “Print” dialog box will open |
| **Actual Result:** | “Print” dialog box will open |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 011 |
| **Unit to test:** | Clicking User Data |
| **Requirements:** | * 3.1.6 Final System Delivery (Visualizing data) |
| **Assumptions:** | When clicking a Member Name, their Data is visualized on the currently selected graph. |
| **Test Data:** | Program\_Teaching\_expanded.csv, Lavi, Shahar |
| **Steps to be executed:** | 1. Load file 2. Click Lavi, Shahar |
| **Expected Result:** | Data is modeled on Pie Chart |
| **Actual Result:** | Data is modeled on Pie Chart |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 012 |
| **Unit to test:** | Clicking Scatter Chart Button |
| **Requirements:** | * 3.1.6 Final System Delivery (Visualizing data) |
| **Assumptions:** | When a member name is selected and the Scatter Chart Button is selected, a scatter chart is produced |
| **Test Data:** | Program\_Teaching\_expanded.csv, Lavi, Shahar |
| **Steps to be executed:** | 1. Load file 2. Click Lavi, Shahar 3. Click Scatter Chart Button |
| **Expected Result:** | Data is modeled on Scatter Chart |
| **Actual Result:** | Data is modeled on Scatter Chart |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 013 |
| **Unit to test:** | Clicking Line Chart Button |
| **Requirements:** | * 3.1.6 Final System Delivery (Visualizing data) |
| **Assumptions:** | When a member name is selected and the Line Chart Button is selected, a line chart is produced |
| **Test Data:** | Program\_Teaching\_expanded.csv, Lavi, Shahar |
| **Steps to be executed:** | 1. Load file 2. Click Lavi, Shahar 3. Click Line Chart Button |
| **Expected Result:** | Data is modeled on Line Chart |
| **Actual Result:** | Data is modeled on Line Chart |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 014 |
| **Unit to test:** | Entering g to h for Filter by: Member Name |
| **Requirements:** | * 3.1.4 Final System Delivery (Applying filters) |
| **Assumptions:** | When characters are entered into the fields for Filtering by: Member Name, all names between the characters are displayed. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Enter g into first box and h into next box |
| **Expected Result:** | All entries between g and h are displayed |
| **Actual Result:** | All entries between g and h are displayed |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 015 |
| **Unit to test:** | Entering g to f for Filter by: Member Name |
| **Requirements:** | * 3.1.4 Final System Delivery (Applying filters) |
| **Assumptions:** | When characters are entered into the fields for Filtering by: Member Name, all names between the characters are displayed. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Enter g into first box and h into next box |
| **Expected Result:** | No entries are displayed |
| **Actual Result:** | No entries are displayed |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 016 |
| **Unit to test:** | Entering 1 to 9 for Filter by: Member Name |
| **Requirements:** | * 3.1.4 Final System Delivery (Applying filters) |
| **Assumptions:** | When characters are entered into the fields for Filtering by: Member Name, all names between the characters are displayed. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Enter g into first box and h into next box |
| **Expected Result:** | All entries are displayed |
| **Actual Result:** | All entries are displayed |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 017 |
| **Unit to test:** | Entering a to z for Filter by: Member Name |
| **Requirements:** | * 3.1.4 Final System Delivery (Applying filters) |
| **Assumptions:** | When characters are entered into the fields for Filtering by: Member Name, all names between the characters are displayed. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Enter g into first box and h into next box |
| **Expected Result:** | All entries are displayed |
| **Actual Result:** | Entries starting with lower case are displayed last |
| **Pass/Fail:** | Fail |
| **Comments:** | As the box parameters are not case sensitize, the program should display the names in alphabetical order regardless of case to avoid confusion. |
| **Solution:** | Convert names to lower case before filtering. |

|  |  |
| --- | --- |
| **Test Case ID:** | 018 |
| **Unit to test:** | Entering 1999 to 2015 for filtering by date |
| **Requirements:** | * 3.1.4 Final System Delivery (Applying filters) |
| **Assumptions:** | When characters are entered into the fields for filtering by date, all names between the dates are displayed. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Enter g into first box and h into next box |
| **Expected Result:** | All entries between the dates are displayed |
| **Actual Result:** | All entries between the dates are displayed |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 019 |
| **Unit to test:** | Entering 2015 to 2014 for filtering by date |
| **Requirements:** | * 3.1.4 Final System Delivery (Applying filters) |
| **Assumptions:** | When characters are entered into the fields for filtering by date, all names between the dates are displayed. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Enter g into first box and h into next box |
| **Expected Result:** | All entries between the dates are displayed, as the last date cannpt be entered |
| **Actual Result:** | All entries between the dates are displayed, as the last date cannpt be entered |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 020 |
| **Unit to test:** | Print Button |
| **Requirements:** | * 3.1.7 Print to File |
| **Assumptions:** | First print button will open printdialog, second print button will print the selected graph. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Select data 3. Press first print button 4. Press print in the dialog box 5. File prints |
| **Expected Result:** | File prints with desired information. |
| **Actual Result:** | File prints with desired information. |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 021 |
| **Unit to test:** | Print Button when no data is selected |
| **Requirements:** | * 3.1.4 Print to File |
| **Assumptions:** | Peachy galaxy will not let you print if you do not have any data selected. |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Press first print button 3. Press print in the dialog box 4. File prints |
| **Expected Result:** | File will not allow user to print |
| **Actual Result:** | File prints blank page. |
| **Pass/Fail:** | Fail |
| **Comments:** | Program should not allow user to print blank page |
| **Solution:** | If there is no data selected, disable to print button. |

|  |  |
| --- | --- |
| **Test Case ID:** | 022 |
| **Unit to test:** | Exporting data |
| **Requirements:** | * 3.1.8 Exporting to PDF |
| **Assumptions:** | Peachy galaxy will export your selected data |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Select data 3. Press export button 4. PDF saves |
| **Expected Result:** | File will export and pdf |
| **Actual Result:** | File exports as pdf |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 023 |
| **Unit to test:** | Exporting data |
| **Requirements:** | * 3.1.8 Exporting to PDF |
| **Assumptions:** | Peachy galaxy will not export to pdf if no data is selected |
| **Test Data:** | Program\_Teaching\_expanded.csv, |
| **Steps to be executed:** | 1. Load file 2. Press export button 3. Export data |
| **Expected Result:** | File will not allow user to export blank file |
| **Actual Result:** | File exports |
| **Pass/Fail:** | Fail |
| **Comments:** | Program should not allow user to export blank file |
| **Solution:** | If there is no data selected, disable the export button. |

|  |  |
| --- | --- |
| **Test Case ID:** | 024 |
| **Unit to test:** | Saving users state |
| **Requirements:** | * Saving users state |
| **Assumptions:** | Peachy galaxy will save users state when prompt accepted |
| **Test Data:** | Program\_Teaching\_expanded.csv |
| **Steps to be executed:** | 1. Load file 2. Select data 3. Exit program, and press yes to save state 4. Start program 5. Data reload into program |
| **Expected Result:** | File currently loaded will be reloaded upon restart |
| **Actual Result:** | File reloads into peachy galaxy |
| **Pass/Fail:** | Pass |
| **Comments:** | Potential bug: Save state does not save current tab, user must switch back to tab which data was loaded into |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 025 |
| **Unit to test:** | Saving users state |
| **Requirements:** | * Saving users state |
| **Assumptions:** | Peachy galaxy will save users only when accepted |
| **Test Data:** | Program\_Teaching\_expanded.csv |
| **Steps to be executed:** | 1. Load file 2. Select data 3. Exit program, and press yes to save state 4. Start program 5. Data reload into program 6. Load different data into program 7. Exit program and decline save 8. Start program, data from step 7 loads |
| **Expected Result:** | Save state will only save when prompt accepted |
| **Actual Result:** | Save state will only save when prompt accepted |
| **Pass/Fail:** | Pass |
| **Comments:** | Potential bug: Save state does not save current tab, user must switch back to tab which data was loaded into |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 026 |
| **Unit to test:** | Clicking Bar Button |
| **Requirements:** | * 3.1.6 Final System Delivery (Visualizing data) |
| **Assumptions:** | When a member name is selected and the Bar Chart Button is selected, a bar chart is produced |
| **Test Data:** | Program\_Teaching\_expanded.csv, Lavi, Shahar |
| **Steps to be executed:** | 1. Load file 2. Click Lavi, Shahar 3. Click Bar Chart Button |
| **Expected Result:** | Data is modeled on Bar Chart |
| **Actual Result:** | Data is modeled on Bar Chart |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 027 |
| **Unit to test:** | Clicking PieChart Button |
| **Requirements:** | * 3.1.6 Final System Delivery (Visualizing data) |
| **Assumptions:** | When a member name is selected and the Pie chart Button is selected, a bar chart is produced |
| **Test Data:** | Program\_Teaching\_expanded.csv, Lavi, Shahar |
| **Steps to be executed:** | 1. Load file 2. Click Lavi, Shahar 3. Click Pie chart Button |
| **Expected Result:** | Data is modeled on Pie Chart |
| **Actual Result:** | Data is modeled on Pie Chart |
| **Pass/Fail:** | Pass |
| **Comments:** | N/A |
| **Solution:** | N/A |

|  |  |
| --- | --- |
| **Test Case ID:** | 028 |
| **Unit to test:** | Column Sorting |
| **Requirements:** | Column Sorting |
| **Assumptions:** | Loaded data can be sorted by their column name. |
| **Test Data:** | Program\_Teaching\_expanded.csv |
| **Steps to be executed:** | 1. Load file 2. Click on column name |
| **Expected Result:** | Data is sorted automatically |
| **Actual Result:** | Data is sorted automatically |
| **Pass/Fail:** | Pass |
| **Comments:** | Column is sorted lexicographically instead of incrementally |
| **Solution:** | N/A |