PicSorter Test Plan

Rayne, Jacob, Paige, Ryan

Test Group	Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
Test Suite 1 - finding directory	TC101	Customer selects valid directory with photo duplicates and click choose	1. Click on Find Directory Button 2. Select valid directory with photo duplicates 3. Click choose	directory path = /Users/Rayne/Desktop/Demo	User should see the groups of duplicates	groups of duplicates are shown	Pass
	TC102	Customer selects valid directory with photo duplicates and click cancel	1. Click on Find Directory Button 2. Select valid directory with duplcates 3. Click cancel	directory path = /Users/Rayne/Desktop/Demo	User should see the home page again	sees home page again	pass
	TC103	Customer selects directory with no photos and click choose	1. Click on Find Directory Button 2. Select directory with no photos 3. Click choose	directory path = /Users/Rayne/Desktop/invalid_demo	user cannot click on choose when no directory is selected	unable to click on choose when no directory is selected	pass
	TC104	Customer selects directory with no photos and click cancel	1. Click on Find Directory Button 2. Select directory with no photos 3. Click choose	directory path = /Users/Rayne/Desktop/invalid_demo	User should see the home page again	sees home page again	pass
	TC105	Customer selects directory with no photos and does nothing	1. Click on Find Directory Button 2. Select directory with no photos	directory path = /Users/Rayne/Desktop/invalid_demo	User should stay on the select directory page	stays on the select directory page until cancel or choose is clicked	pass

	TC106	Customer selects valid directory with photo duplicates and does nothing	1. Click on Find Directory Button 2. Select valid directory with photo duplicates	directory path = /Users/Rayne/Desktop/Demo	User should stay on the select directory page	stays on the select directory page until cancel or choose is clicked	pass
	TC107	Customer selects directory with photos but no duplicates and click choose	1. Click on Find Directory Button 2. Select directory with no duplicates 3. Click choose	directory path = /Users/Rayne/Desktop/invalid_demo	User should see the home page again	pop up says "no duplicates are found" and sees home page again	pass
	TC108	Customer selects directory with photos but no duplicates and does nothing	1. Click on Find Directory Button 2. Select directory with photos and no duplicates	directory path = /Users/Rayne/Desktop/invalid_demo	User should stay on the select directory page	stays on the select directory page until cancel or choose is clicked	pass
	TC109	Customer selects directory with no photos and click cancel	1. Click on Find Directory Button 2. Select directory with no duplicates 3. Click choose	directory path = /Users/Rayne/Desktop/invalid_demo	User should see the home page again	sees home page again	pass
Test Suite 2 - multiple selection and deletion	TC201	User selects multiple photos and confirms delete	1. Click on multiple photos. 2. Click the delete button. 3. Click yes.	Photo1 path = Users/paigehanssen/Desktop/PicSorterTest/more photos/madeira_edit; Photo2 path = Users/paigehanssen/Desktop/PicSorterTest/more	Photos should be deleted permanently from the group and the window pane.	Photos removed from computer and from the frame.	pass
	TC202	User selects multiple photos and cancels delete	1. Click on multiple photos. 2. Click the Move to Trash button. 3. Click no.	Photo1 path = Users/paigehanssen/Desktop/PicSorterTest; Photo2 path = Users/paigehanssen/Desktop/PicSorterTest/	Photos should not be deleted.	Photos are not deleted.	pass

	TC203	User selects multiple photos and exits the frame	1. Select photos. 2. Click the "X" in top left corner to close frame.	Photo1 path = Users/paigehanssen/Desktop/PicSorterTest; Photo2 path = Users/paigehanssen/Desktop/PicSorterTest/	Photos should remain in the folder and duplicate group. Duplicate viewing frame should close.	Photos remain, frame is closed properly.	pass
	TC204	User does not select any photos and tries to delete them	1. Click the delete button without any selected photos.	Selected photos: none	Nothing should happen when delete button is clicked.	Nothing happens when delete button is clicked.	pass
	TC205	User selects multiple photos, deselects, then tries to delete	1. Select photos. 2. Deselect all photos. 3. Click delete button.	Selected photos: none	Delete button should not be activated and nothing should happen when clicked.	Nothing happens when delete button is clicked.	pass
	TC206	User selects, deselects photos and reselects, then clicks delete	1. Select photos. 2. Deselect all photos. 3. Select one or more photos. 4. Delete selected photos.	Photo1 path = Users/paigehanssen/Desktop/PicSorterTest/more photos/PSlogo; Photo2 path = Users/paigehanssen/Desktop/PicSorterTest/more photos/PSlogo copy	Most recently selected photos should be deleted.	Only the selected photos are deleted.	pass
	TC207	User selects, deselects photos and reselects, then clicks and cancels delete	1. Select photos. 2. Deselect all photos. 3. Select one or more photos. 4. Click Move to Trash button. 5. Click no.	Photo1 path = Users/paigehanssen/Desktop/PicSorterTest/more photos/PSlogo; Photo2 path = Users/paigehanssen/Desktop/PicSorterTest/more photos/PSlogo copy	Most recently selected photos should be deleted.	No photos are deleted.	pass
	TC208	User selects one photo and confirms delete	1. Select one photo. 2. Click the delete button. 3. Click yes	Photo path = Users/paigehanssen/Desktop/PicSorterTest/more photos/IMG_1061 copy 2; Destination: deleted permanently	Photo should be deleted from the original photo and the window pane.	Photos are deleted from the window and from the group of duplicates.	pass
	TC209	User selects one photo and cancels delete	1. Select one photo. 2. Click the delete button. 3. Click no.	Photo path = Users/paigehanssen/Desktop/PicSorterTest/more photos/IMG_1061	Photo should not be deleted.	Photo is not deleted	pass

	TC210	User selects one photo and exits the frame	1. Select one photo. 2. Click the "X" in top left corner to close frame.	Selected photos: none	Photo should remain in the folder and duplicate group. Duplicate viewing frame should close.	Photos remain in the folder and in the group, frame is closed.	pass
	TC211	User selects one photo, deselects, then tries to delete	1. Select one photo. 2. Deselect photo. 3. Click delete button.	Selected photos: none	Nothing should happen when delete button is clicked.	Nothing happens when delete button is clicked.	pass
	TC212	User selects one photo, deselects photo and reselects, then clicks delete	1. Select one photo. 2. Deselect photo. 3. Select one photo. 4. Click delete button. 5. Click yes	Photo path = Users/paigehanssen/Desktop/PicSorterTest/more photos/neature; Destination: permanently deleted.	Most recently selected photo should be deleted.	Most recently selected photo is deleted.	pass
	TC213	User selects one photo, deselects photo and reselects, then clicks and cancels delete	1. Select one photo. 2. Deselect photo. 3. Select one photo. 4. Click delete. 5. Click no	Photo path = Users/paigehanssen/Desktop/PicSorterTest/more photos/n85re.So.79 copy	No photos should be deleted.	No photos are deleted.	pass
	TC214	User selects all photos, clicks the delete button and confirms	1. Select all photos. 2. Click the delete button. 3. Click yes	Photo1 path = Users/paigehanssen/Desktop/PicSorterTest copy/more photos/n85re.So.79 copy; Photo2 path = Users/paigehanssen/Desktop/PicSorterTest copy/more photos/n85re.So.79; Destination:	Photos should be deleted from the original photo and the window pane, frame should close and group should be removed.	Photos are deleted from group, frame closes, group is removed.	pass

	TC215	User selects all photos and clicks the delete button and cancels	1. Select all photos. 2. Click the delete button. 3. Click no	Photo1 path = Users/paigehanssen/Desktop/PicSorterTest copy/more photos/madeira_edit copy; Photo2 path = Users/paigehanssen/Desktop/PicSorterTest copy/more photos/madeira_edit copy 2; Photo3 path =	Photos should not be deleted.	No photos are deleted.	pass
	TC216	User selects all photos and exits the frame	1. Select one photo. 2. Click the "X" in top left corner to close frame.	Photo1 path = Users/paigehanssen/Desktop/PicSorterTest copy/more photos/madeira_edit copy; Photo2 path = Users/paigehanssen/Desktop/PicSorterTest copy/more photos/madeira_edit copy 2; Photo3 path =	Photo should remain in the folder and duplicate group. Duplicate viewing frame should close.	Photos remain in the folder and in the group, frame is closed.	pass
	TC217	User selects all photos, deselects, then tries to delete	1. Select one photo. 2. Deselect photo. 3. Click delete button.	Selected photos: none	Nothing should happen when delete button is clicked.	Nothing happens when delete button is clicked.	pass
	TC218	User selects all photos, deselects photos and reselects, then clicks and confirms delete	1. Select one photo. 2. Deselect photo. 3. Select one photo. 4. Click delete button. 5. Click yes	Photo1 path = Users/paigehanssen/Desktop/PicSorterTest copy/more photos/madeira_edit copy; Photo2 path = Users/paigehanssen/Desktop/PicSorterTest copy/more photos/madeira_edit copy 2; Photo3 path = Users/paigehanssen/Desktop/PicS	All photos should be deleted, frame should close and group of duplicates should be removed.	All duplicates are deleted, frame is closed, that duplicate group is no longer displayed.	pass

	TC219	User selects all photos, deselects photos and reselects, then clicks and cancels delete.	1. Select one photo. 2. Deselect photo. 3. Select one photo. 4. Click delete button. 5. Click no.	Photo1 path = Users/paigehanssen/Desktop/PicSorterTest copy/neature; Photo2 path = Users/paigehanssen/Desktop/PicSorterTest copy/neature copy; Photo3 path =	No photos should be deleted.	No photos are deleted.	pass
Test Suite 3 - multiple selection and moving selected duplicates to a new location	TC301	User selects a single photo and tries to move it to an existing folder	1. Select a single photo 2. Click the move button 3. Select an existing folder 4. Click choose	Destination Directory path=Users/jacobtower/PicSorterTest Photo path=Users/jacobtower/photos/neature.png	User should successfully move photo to selected location, remove moved photo from frame displaying all duplicates	Photo was moved to the correct directory and frame was updated to remove the correct image	pass
	TC302	User selects a group of photos and tries to move them to an existing folder	1. Select a group of photos 2. Click the move button 3. Select an existing folder 4. Click Choose	Destination Directory path=Users/jacobtower/PicSorterTest photo1 path=Users/jacobtower/photos/stock1.jpg photo2 path=Users/jacobtower/photos/stock1 copy.jpg	User should successfully move photos to selected location, all selected photos should be removed from frame displaying all duplicate images	All selected photos were moved to the correct directory and removed from the display of all duplicate images	pass
	TC303	User selects a single photo and tries to move it to an invalid location (not a folder)	1. Select a single photo 2. Click the move button 3. Select a file that is not a folder 4. Select Choose	Destination File Path=Users/jacobtower/eclipse-workspace/PicSorter/src/Home.java Photo path=Users/jacobtower/photos/neature.png	User should not be able to select the non-folder file	Non-folder file could not be selected	pass

	TC304	User selects a group of photos and tries to move them to an invalid location (not a folder)	1. Select a single photo 2. Click the move button 3. Select a file that is not a folder 4. Select Choose	Destination File Name = Home.java	User should not be able to select the non-folder file	Non-folder file could not be selected	pass
	TC305	User selects a single photo, tries to move it, and hits cancel	1. Select a single photo 2. Click the move button 3. Click the cancel button	Photo path=Users/jacobtower/photos/neature.png	Finder should close and return to the selection screen	Finder closed and returned to selection screen	pass
	TC306	User selects a group of photos, tries to move them, and hits cancel	1. Select a group of photos 2. Click the move button 3. Click the cancel button	photo1 path = Users/jacobtower/photos/stock1.jpg photo2 path= Users/jacobtower/photos/stock1 copy.jpg	Finder should close and return to the selection screen	Finder closed and returned to selection screen	pass
	TC307	User selects a single photo and closes the window while selecting a destination	1. Select a single photo 2. Click the move button 3. Click the close window button	Photo path=Users/jacobtower/photos/neature.png	User should not be able to close window, but should be able to close the finder	Finder closed and returned to selection screen	pass
	TC308	User selects a group of photos and closes the window while selecting a destination	1. Select a single photo 2. Click the move button 3. Click the close window button	photo1 path = Users/jacobtower/photos/stock1.jpg photo2 path= Users/jacobtower/photos/stock1 copy.jpg	User should not be able to close window, but should be able to close the finder	Finder closed and returned to selection screen	pass
	TC309	User does not select any photos and tries to move them	1. Click the move photos button	Photo directory=Users/jacobtower/photos	Button should not perform any function	User can click on button but nothing occurs	pass

	TC310	User selects all photos in a duplicate set and moves them	1. Select all images in a duplicate set 2. Click the move buttton 3. Select the destination folder 4. Click choose	Photo directory=Users/jacobtower/photos photo1 path = Users/jacobtower/photos/stock1.jpg photo2 path= Users/jacobtower/photos/stock1 copy.jpg Source Directory=Users/jacobtower/PicSorterTest	All photos selected should have the new location. The frame should close and the duplicate set should be removed from the display of all sets.	All selected photos were moved to the correct directory and the window displaying the duplicates was closed. The display of all duplicate sets was adjusted to no longer contain that set.	pass
Test Suite 4 - system performance	TC401	Load 100 images into the hash map	**1.** Create folder test_folder **2.** Add 100 images to test_folder **3.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **4.** Record the time it takes to load (the output of the test program)	Folders containing images: Directory - test_folder	Load time < 1 sec.	10.189260435 sec.	Fail
	TC402	Load 250 images into the hash map	**1.** Add 150 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time < 1 sec.	24.378751921 sec.	Fail

	TC403	Load 500 images into the hash map	**1.** Add 250 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time < 1 sec.	50.464527989 sec.	Fail
	TC404	Load 1000 images into the hash map	**1.** Add 500 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time = 1 sec.	108.459579726 sec.	Fail
	TC405	Load 2500 images into the hash map	**1.** Add 1500 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time = 1.5 sec.	296.351140724 sec.	Fail
	TC406	Load 5000 images into the hash map	**1.** Add 2500 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time = 3 sec.	644.316035517 sec.	Fail

	TC407	Load 10000 images into the hash map	**1.** Add 5000 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time = 5 sec.	1210.539088806 sec.	Fail
	TC408	Load 1000 images in 10 sub directories into the hash map	**1.** Delete 9000 images in test_folder **2.** Create 10 folders in test_folder called subfolder<n> **3.** Add 100 images to each subfolder **4.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **5.** Record the time it	Folders containing images: Directory - test_folder; Sub-directories: sub_folder_n (where n is the number of the folder)	Load time = 1 sec.	128.071140724 sec.	Fail
	TC409	Load 1000 images in 25 sub directories into the hash map	**1.** Create 15 more folders in test_folder called subfolder<n> **2.** Move 40 images to each subfolder from those subfolders containing 100 so that every folder has 40 images **3.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **4.** Record the time it takes to load (the	Folders containing images: Directory - test_folder; Sub-directories: sub_folder_n (where n is the number of the folder)	Load time = 1.25 sec.	133.268486376 sec.	Fail

	TC410	Load 1000 images in 50 sub directories into the hash map	**1.** Create 25 more folders in test_folder called subfolder<n> **2.** Move 25 images to each subfolder from those subfolders containing 40 so that every folder has 25 images **3.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **4.** Record the time it takes to load (the	Folders containing images: Directory - test_folder; Sub-directories: sub_folder_n (where n is the number of the folder)	Load time = 1.5 sec.	125.889764098 sec.	Fail
	TC411	Load 1000 images in 100 sub directories into the hash map	**1.** Create 50 more folders in test_folder called subfolder<n> **2.** Move 10 images to each subfolder from those subfolders containing 25 so that every folder has 10 images **3.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **4.** Record the time it takes to load (the	Folders containing images: Directory - test_folder; Sub-directories: sub_folder_n (where n is the number of the folder)	Load time = 2 sec.	124.934012353 sec.	Fail

	TC412	Load 1000 images in 10 sub directories and 1 sub sub directory each (for a total of 20 directories) into the hash map	**1.** Create 10 folders in test_folder called subfolder<n> **2.** Move 100 images to each subfolder **3.** Create a subsubfolder in each of these subfolders **4.** Move 50 images in each subfolder into the subsubfolder **5.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **6.** Record the time it	Folders containing images: Directory - test_folder; Sub-directories: sub_folder_n (where n is the number of the folder); Sub-sub-directories: sub_sub_folder	Load time = 5 sec.	125.428625453 sec.	Fail
Test Suite 5 - test system performance using file size hashing instead of dHash hashing	TC501	Load 100 images into the hash map	**1.** Create folder test_folder **2.** Add 100 images to test_folder **3.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **4.** Record the time it takes to load (the output of the test program)	Folders containing images: Directory - test_folder	Load time < 1 sec.	0.002092228 sec.	Pass
	TC502	Load 250 images into the hash map	**1.** Add 150 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time < 1 sec.	0.005128078 sec.	Pass

	TC503	Load 500 images into the hash map	**1.** Add 250 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time < 1 sec.	0.009097533 sec.	Pass
	TC504	Load 1000 images into the hash map	**1.** Add 500 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time = 1 sec.	0.0126495 sec.	Pass
	TC505	Load 2500 images into the hash map	**1.** Add 1500 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time = 1.5 sec.	0.037860121 sec.	Pass
	TC506	Load 5000 images into the hash map	**1.** Add 2500 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time = 3 sec.	0.083593799 sec.	Pass

	TC507	Load 10000 images into the hash map	**1.** Add 5000 images to test_folder **2.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **3.** Record the time it takes to load (the	Folders containing images: Directory - test_folder	Load time = 5 sec.	0.116315884 sec.	Pass
	TC508	Load 1000 images in 10 sub directories into the hash map	**1.** Delete 9000 images in test_folder **2.** Create 10 folders in test_folder called subfolder<n> **3.** Add 100 images to each subfolder **4.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **5.** Record the time it	Folders containing images: Directory - test_folder; Sub-directories: sub_folder_n (where n is the number of the folder)	Load time = 1 sec.	0.012543668 sec.	Pass
	TC509	Load 1000 images in 25 sub directories into the hash map	**1.** Create 15 more folders in test_folder called subfolder<n> **2.** Move 40 images to each subfolder from those subfolders containing 100 so that every folder has 40 images **3.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **4.** Record the time it takes to load (the	Folders containing images: Directory - test_folder; Sub-directories: sub_folder_n (where n is the number of the folder)	Load time = 1.25 sec.	0.014243881 sec.	Pass

	TC510	Load 1000 images in 50 sub directories into the hash map	**1.** Create 25 more folders in test_folder called subfolder<n> **2.** Move 25 images to each subfolder from those subfolders containing 40 so that every folder has 25 images **3.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **4.** Record the time it takes to load (the	Folders containing images: Directory - test_folder; Sub-directories: sub_folder_n (where n is the number of the folder)	Load time = 1.5 sec.	0.013325207 sec.	Pass
	TC511	Load 1000 images in 100 sub directories into the hash map	**1.** Create 50 more folders in test_folder called subfolder<n> **2.** Move 10 images to each subfolder from those subfolders containing 25 so that every folder has 10 images **3.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **4.** Record the time it takes to load (the	Folders containing images: Directory - test_folder; Sub-directories: sub_folder_n (where n is the number of the folder)	Load time = 2 sec.	0.014003961 sec.	Pass

	TC512	Load 1000 images in 10 sub directories and 1 sub sub directory each (for a total of 20 directories) into the hash map	**1.** Create 10 folders in test_folder called subfolder<n> **2.** Move 100 images to each subfolder **3.** Create a subsubfolder in each of these subfolders **4.** Move 50 images in each subfolder into the subsubfolder **5.** Compile "javac HashTest.java" and run "java HashTest <filepath>/test_folder " **6.** Record the time it	Folders containing images: Directory - test_folder; Sub-directories: sub_folder_n (where n is the number of the folder); Sub-sub-directories: sub_sub_folder	Load time = 5 sec.	0.012731781 sec.	Pass
	TC513	Load 50000 images in 750 different folders into the hash map	**This test was run on my pictures folder to test the upper limit. It is not an official test and is only put here for personal interest as to the capabilities**			2.482117161 sec.	