**Project 1: Ticket Manager Closed-Box Test Plan**

**Document Author(s): Darien Gillespie and Marcus Ribeiro**

**Date: 9/26/2022**

**Introduction**

The program will be started by running the TicketManagerUI class as a Java application. This will open the interactable window that the user can use to load, view, create, and save tickets. The following 8 tests will cover all 8 case uses represented in the project requirements to ensure that they are all working.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Results** | **Actual Results** |
| Test 1: Ticket File Interactions | **Preconditions:** *None*   1. Select “New” from the file menu. 2. Create a text file with the name “ticket.txt” and the contents as follows: \*1#New#Incident#Help#Darien#Inquiry#Urgent## -Sample Note -Sample Note 2 3. Select “Load” from the file menu. 4. Select the previously created file in the file browser. 5. Select “Open” from the file menu. 6. Select the previously created file in the file browser. 7. Select “Quit” from the file menu. 8. Type in “newfile.txt” as the file name in the file browser. 9. Click “Save.” 10. Create a second text file with the name “ticket2.txt” and the contents as follows: This is not a valid ticket. 11. Select “Load” from the file menu. 12. Select “ticket2.txt”. 13. Click Ok on the popup window. 14. Close the application. | * After Step 1, a new empty list of tickets should be created. * After Step 3, a file browser should open. * After Step 4, the loaded ticket show display in the ticket list. * After Step 5, a file browser should open. * After Step 7, a file browser should open. * After Step 9, the open file should be saved to newfile.txt and the application should close. * After Step 12, a window should pop up with the error “Unable to load file.” | * After Step 9, the system created a new file menu, added a ticket that we generated in a text file, read it in, and saved the file as we quit the menu. * After Step 14, ticket2.txt was not able to be loaded in a separate file menu as the file is invalid and then the application was closed. |
| Test 2: Ticket List | **Preconditions:** *User has selected a ticket list to display*   1. Click “Add New Ticket” in the file menu. 2. Fill the fields with the following information: Ticket Type: Incident Subject: Test Caller ID: Darien Category: Software Priority: Urgent Note: This is a test ticket being added. 3. Click “Add Ticket.” 4. Repeat Steps 1-3, but replacing the Category with: Category: Invalid 5. Click “Return” 6. Click “Edit Ticket.” 7. Select the ticket created by the first run through from the list. 8. Click “Edit Ticket.” 9. Click “Working.” 10. Select the “Resolved” ticket type filter. 11. Select “Show All Tickets.” 12. Select the ticket from the list. 13. Click “Delete Ticket.” 14. Close the application. | * After Step 1, a window should open with fields for the data for a ticket * After Step 3, the user should be returned to the ticket list with the new ticket listed. * After Step 4, a dialog should open saying “Ticket cannot be created.” * After Step 5, the user should be returned to the ticket list with only one ticket listed. * After Step 6, * After Step 8, the user should be brought to a page displaying the ticket information and buttons for the various states. * After Step 10, no tickets should be displayed. * After Step 11, the ticket should be visible again. * After Step 13, the ticket should be removed from the list. | * Steps 1-3 worked just as expected, with the ticket being added to the system. * Steps 4-5 does not produce invalid as expected since this GUI does not let one enter in an invalid category. * Steps 6-8 work as expected, as the ticket we first created in Test 2 is being put in Edit mode. * Step 9 does not work as there is no working functionality. But you can write a work note and investigate. So, I did this with an ownerID = “Marcus,” a work note of “work note test,” and I chose a cancelation code of “Duplicate.” * Step 10 was not possible as one can only select between Incident and Request filters. * Step 11 worked as expected, showing all tickets. * Steps 12-14 worked perfectly as the ticket was deleted and the application was closed. |
| Test 3: New State | **Precondition:** *The user has created three tickets that are in the “New” state and selected the first one.*   1. Click “Edit Ticket.” 2. Click “Investigate.” 3. Click “OK.” 4. Enter “Darien” in the ID field. 5. Click “Investigate” 6. Click “OK.” 7. Enter “Looking into this ticket” in the note field. 8. Click “Investigate.” 9. Select the second ticket from the list. 10. Click “Edit Ticket.” 11. Click “Cancel.” 12. Click “OK.” 13. Enter “This ticket is invalid” in the note field. 14. Click “Cancel.” 15. Select the third ticket from the list. 16. Click “Edit Ticket.” 17. Click “Return.” 18. Close the application. | * After Step 2, a dialog should open stating “Invalid owner id.” * After Step 5, a dialog should open stating “Invalid command.” * After Step 8, the user should be returned to the ticket list. * After Step 11, a dialog should open stating “Invalid command.” * After Step 14, the user should be returned to the ticket list with the state of the ticket updated * After Step 17, the user should be returned to the ticket list with nothing changed. | * Steps 1-2 were performed and so the first ticket was put in edit mode and investigate returned an “invalid owner id” popup as expected. * Steps 3-5 resulted in “Darien” being entered in the ID field and an “Invalid command” popup appearing. * Steps 6-8 made the ticket investigation start as a note was added. * Steps 9-12 resulted in an invalid command popup appearing as expected since the user tried to cancel without entering appropriate information. * Step 13 assumed that the edit ticket window would stay open, but this had to be re-opened. After re-opening the edit window, I continued following the directions. * Steps 13-14 resulted in the ticket being canceled with a note. * Steps 15-18 worked as expected, clicking on a new ticket and being able to return the ticket list again, and finally leaving the application. |
| Test 4: Working State | **Preconditions:** *The user has created three tickets that are in the “Working” state and selected the first one.*   1. Click “Edit Ticket.” 2. Click “Awaiting Change” in the combo box. 3. Click “Feedback.” 4. Click “OK.” 5. Enter “Need response to work” in the Note field. 6. Click “Feedback.” 7. Select the second ticket. 8. Click “Edit Ticket.” 9. Select “Solved” from the combo box. 10. Click “Resolved.” 11. Click “OK.” 12. Enter “Solution implemented” in the Note field. 13. Click “Resolved.” 14. Select the third ticket. 15. Click “Edit Ticket.” 16. Select “Inappropriate” from the combo box. 17. Click “Cancel.” 18. Click “OK.” 19. Enter “Not enough information” in the Note field. 20. Click “Cancel.” 21. Close the application. | * After Step 3, a dialog should pop up stating “Invalid command.” * After Step 6, the user should be returned to the ticket list and the ticket’s state should be updated. * After Step 10, a dialog should pop up stating “Invalid command.” * After Step 13, the user should be returned to the ticket list and the ticket’s state should be updated. * After Step 17, a dialog should pop up stating “Invalid command.” * After Step 20, the user should be returned to the ticket list and the ticket’s state should be updated. | * Steps 1-4 lets you edit the ticket as expected and set it to Awaiting Change, failing at clicking Feedback as expected. However, it returns you to the main ticket window, so I will re-input Step 1-2’s changes and continue from Step 5. As such, Steps 5-6 work as suggested. * Steps 7-11 create an invalid command popup as expected but also return the user to the ticket menu, so I will input changes prior to clicking resolved and continue from here. * Now Steps 12-13 work as expected. * Steps 14-21 work as expected but the information needs to be re-inputted as the error puts you back at the ticket list. |
| Test 5: Feedback State | **Preconditions:** *The user has created four tickets that are in the “Feedback” state and selected the first one.*   1. Click “Edit Ticket.” 2. Click “Reopen.” 3. Click “OK.” 4. Enter “More work needed” in the Note field. 5. Click “Reopen.” 6. Select the second ticket from the list. 7. Click “Edit Ticket.” 8. Select “Completed” as the resolution code from the combo box if it’s a Request or “Solved” if it is an Incident. 9. Click “Resolve.” 10. Click “OK.” 11. Enter “Solution committed.” In the Note field. 12. Click “Resolve.” 13. Select the third ticket from the list. 14. Click “Edit Ticket.” 15. Select “Invalid” from the combo box. 16. Click “Cancel.” 17. Click “OK.” 18. Enter “Outdated” in the Note field. 19. Click “Cancel.” 20. Select the fourth ticket from the list. 21. Click “Edit Ticket.” 22. Click “Return.” 23. Close the application. | * After Step 3, a dialog should open stating “Invalid command.” * After Step 5, the user should be returned to the ticket list with the ticket’s state updated to Working. * After Step 9, a dialog should open stating “Invalid command.” * After Step 12, the user should be returned to the ticket list and the ticket’s state should be updated to Resolved. * After Step 16, a dialog should pop up stating “Invalid command.” * After Step 19, the user should be returned to the ticket list and the ticket’s state should be updated to Closed. * After Step 22, the user should be returned to the ticket list with no changes. | * Steps 1-3 work just as expected. * Step 4 can only run if you redo step 1 first. Accordingly, steps 4-5 work as expected. * Steps 5-12 work just as expected if you redo steps 7-8 after 10. * Steps 13-19 work as expected except the ticket does not change to Closed but stays as a Request. Also, since there is no invalid option, I reversed the advice of Step 8. Additionally, I had to reinput data after the error pop up redirected me out of the editing window. * Steps 20-23 worked as expected. |
| Test 6: Resolved State | **Preconditions:** *The user has created four tickets that are in the “Resolved” state and selected the first one.*   1. Click “Edit Ticket.” 2. Select “Awaiting Change” from the combo box. 3. Click “Feedback.” 4. Click “OK.” 5. Enter “Need confirmation of resolution” in the Note field 6. Click “Feedback.” 7. Select the second ticket. 8. Click “Edit Ticket.” 9. Click “Reopen.” 10. Click “OK.” 11. Enter “Error persists” in the Note field. 12. Click “Reopen.” 13. Select the third ticket. 14. Click “Edit Ticket.” 15. Click “Confirm.” 16. Click “OK.” 17. Enter “Solution works” in the Note field. 18. Click “Confirm.” 19. Select the fourth ticket. 20. Click “Edit Ticket.” 21. Click “Return.” 22. Close the application. | * After Step 3, a dialog should pop up stating “Invalid command.” * After Step 6, the user should be returned to the ticket list with the ticket’s state updated to “Feedback.” * After Step 9, a dialog should pop up stating “Invalid command.” * After Step 12, the user should be returned to the ticket list with the ticket’s state updated to “Working.” * After Step 15, a dialog should pop up stating “Invalid command.” * After Step 18, the user should be returned to the ticket list with the ticket’s state updated to “Closed.” * After Step 21, the user should be returned to the ticket list with no changes. | * Steps 1-6 worked as expected, except for the recurring added step of re-inputting pre-error dialogue data in the editing window. * Steps 7-12 work as expected, except for the recurring added step of re-inputting pre-error dialogue data in the editing window. * Steps 13-18 work as expected, except for the recurring added step of re-inputting pre-error dialogue data in the editing window. * Steps 19-22 worked as expected. |
| Test 7: Closed State | **Preconditions:** *The user has created two tickets that are in the “Closed” state and selected the first one.*   1. Click “Edit Ticket.” 2. Click “Reopen.” 3. Click “OK.” 4. Enter “Fix no longer works” in the Note field. 5. Click “Reopen.” 6. Select the second ticket. 7. Click “Edit Ticket.” 8. Click “Return.” 9. Close the application. | * After Step 2, a dialog should pop up stating “Invalid command.” * After Step 5, the user should be returned to the ticket list with the ticket’s state updated to “Working.” * After Step 8, the user should be returned to the ticket list with no changes. | * Steps 1-5 work as expected, except for the recurring added step of re-inputting pre-error dialogue data in the editing window. * Steps 6-9 work just as expected. |
| Test 8: Cancelled State | **Preconditions:** *The user has created a ticket that is in the “Canceled” state and selected the first one.*   1. Click “Edit Ticket.” 2. Click “Return.” | * After Step 2, the user should be returned to the ticket list with no changes. | * Steps 1-2 work just as expected. |

**Document Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Change Description** |
| **9/26/2022** | Darien Gillespie | * Added Tests 1-4 |
| **9/27/2022** | Darien Gillespie | * Added Tests 5-8 |
| **10/24/2022** | Marcus Ribeiro | * Added Actual Results |