**<TicketManager> Closed-Box Test Plan**

**Document Author(s): Michael Martinez**

**Date: 9/27/2022**

**Introduction**

There are eight system tests. Each test is linked to a valid scenario or sub-flow for each use case. In addition, there is a [TicketManager test file](https://bowdoin-my.sharepoint.com/:t:/g/personal/mmartinez_bowdoin_edu/ERcvakFasFFFsYYNW9tAfLwB5m5qHPUI8lgwCEKcTttX_g?e=1wbxg6) provided that is required to run each test.

To run the tests:

1. Right click on TicketManagerUI class in the Package Explorer
2. Select Run As > Java Application

Note that while all tests except for Test 1 end with the statement to Close GUI, you do not need to re-launch the GUI each time as long as there are no failures. If a test fails, “ticketmanagertest.txt” must be loaded into the program each time.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Results** | **Actual Results** |
| **Test 1:**  Load valid ticket file (Michael) | **Preconditions:** Download the provided “[ticketmanagertest.txt](https://bowdoin-my.sharepoint.com/:t:/g/personal/mmartinez_bowdoin_edu/ERcvakFasFFFsYYNW9tAfLwB5m5qHPUI8lgwCEKcTttX_g?e=ZWodVS)”   1. Run TicketManagerUI 2. Click **Load** 3. Browse for the provided “[ticketmanagertest.txt](file:///Users/michaelmartinez/Library/Mobile%20Documents/com~apple~TextEdit/Documents/ticketmanagertest.txt)” file in your file manager & open it 4. *Check Results* | After Step 1: TicketManagerUI loads  After Step 3: A new ticket list should appear with six tickets:  “1. CodeRunner down…  2. Workshop account…  3. Add Gradescope plugin to Canvas…  4. Lights not working in Searles 223…  5. New VM…  6. Pizza…” |  |
| **Test 2:**  Filter ticket list  (Michael) | **Preconditions:** Test 1 has passed.   1. Select the test ticket list to display 2. Click **Filter** 3. Either type “Incident” into the combo box or select it in the dropdown menu 4. *Check results* 5. Click **Filter** 6. Either type “Request” into the combo box or select it in the dropdown menu 7. *Check results* 8. Click “Show All Tickets” 9. *Check results* 10. Close GUI | After Step 1: The ticket id, ticket type, state, subject, category, and priority are displayed for all tickets in the system.  After Step 3: Each ticket displayed should be of type, “Incident”  After Step 6: Each ticket displayed should be of type, “Request”  After Step 8: Every ticket should be displayed again |  |
| **Test 3:**  Assign a New ticket to an owner  (Michael) | **Preconditions:** Test 1 has passed.   1. Select the first ticket labeled “CodeRunner down” 2. Click **Edit** 3. Enter “martinez” in the owner id field 4. Enter “Assigned to martinez” as a note for the ticket 5. Click **Investigate** 6. *Check results* 7. Close GUI | * Ticket’s state should now be “Working” * Ticket’s ownerId should now be “martinez” * The new note should be saved with the ticket * User should be returned to ticket list view * The ticket’s listing reflects the updated state in the ticket list |  |
| **Test 4:**  Request feedback for a Working ticket  (Michael) | **Preconditions:** Test 1 has passed.   1. Select the ticket labeled “Workshop account” 2. Click **Edit** 3. Select the feedback code, “Awaiting Provider,” in the combo box 4. Enter “requested feedback: awaiting provider” as a note for the ticket 5. Click **OK** 6. *Check Results* 7. Close GUI | * Ticket’s state should now be “Feedback” * Ticket’s feedback code should now be “Awaiting Provider” * The new note should be saved with the ticket * User should be returned to ticket list view * The ticket’s listings reflect the updated state in the ticket list |  |
| **Test 5:**  Reopen a Feedback ticket  (Michael) | **Preconditions:** Test 1 has passed.   1. Select the ticket labeled “Add Gradescope plugin to Canvas” 2. Click **Edit** 3. Enter “Reopened ticket” as a note for the ticket 4. Click **Reopen** 5. *Check Results* 6. Close GUI | * Ticket’s state should now be “Working” * Ticket should not have the feedback code, “Awaiting Provider,” anymore * The new note should be saved with the ticket * User should be returned to ticket list view * The ticket’s listings reflect the updated state in the ticket list |  |
| **Test 6:**  Confirm a Resolved ticket as resolved  (Michael) | **Preconditions:** Test 1 has passed.   1. Select the ticket labeled “Lights not working in Searles 223” 2. Click **Edit** 3. Enter “Incident handled” as a note for the ticket 4. Click **Confirm** 5. *Check Results* 6. Close GUI | * Ticket’s state should now be “Closed” * Ticket should not have the resolution code, “Workaround,” anymore * The new note should be saved with the ticket * User should be returned to ticket list view * The ticket’s listings reflect the updated state in the ticket list |  |
| **Test 7:**  Reopen a Closed ticket  (Michael) | **Preconditions:** Test 1 has passed.   1. Select the ticket labeled “New VM” 2. Click **Edit** 3. Enter “Reopened ticket” as a note for the ticket 4. Click **Reopen** 5. *Check Results* 6. Close GUI | * Ticket’s state should now be “Working” * Ticket should not have the resolution code, “Completed,” anymore * The new note should be saved with the ticket * User should be returned to ticket list view * The ticket’s listings reflect the updated state in the ticket list |  |
| **Test 8:**  Return to the ticket list with no change from a Canceled ticket  (Michael) | **Preconditions:** Test 1 has passed.   1. Select the ticket labeled “Pizza” 2. Click **Edit** 3. Click **Return** 4. *Check Results* 5. Close GUI | * Ticket’s attributes should remain the same * User should be returned to ticket list view * The ticket’s listings reflect the unchanged state in the ticket list |  |

**Document Revision History**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Change Description** |
| **9/27/2022** | Michael Martinez | * Created eight tests for valid scenarios, one for each use case |