**System Test Plan**

|  |
| --- |
| **Instructions.** In this section, you must provide your system test plan with at least 5 test cases. **If you want to provide more than 5 test cases, add an appendix at the end of this document with the 6th, 7th, etc. test cases so that page numbers for all sections match the required template!**  Make sure:   * You provide your sample test data * Test IDs are uniquely identified and descriptive * Test descriptions are fully specified with complete inputs, specific values, and preconditions   + Be sure to provide SPECIFIC INPUTs and VALUEs so that your test cases are repeatable * Expected results are fully specified with specific output values * All tests cover scenarios based on the problem statement * All tests cover unique scenarios for the system * All strategies for system testing are demonstrated in the tests (testing equivalence classes, testing boundary values, testing exceptions/unexpected inputs) |

**Test Data:**

For our system test cases, we will use the following text files named rooms1.txt, logs1.txt, rooms2.txt, logs2.txt

*rooms2.txt:*

ROOM\_ID,LENGTH,WIDTH

*logs2.txt:*

TIMESTAMP,ROOM\_ID,PERCENT\_CLEANED

*rooms1.txt:*

ROOM\_ID,LENGTH,WIDTH

Main Bedroom,62,65

Main Bathroom,5,5

Living Room,15,15

Kitchen,10,12

Dining Room,8,8

Entryway,5,5

*logs1.txt*

TIMESTAMP,ROOM\_ID,PERCENT\_CLEANED

09/13/2023 20:08:05,Dining Room,100

09/13/2023 19:02:55,Kitchen,98

09/11/2023 20:21:19,Dining Room,80

09/11/2023 19:09:10,Kitchen,79

09/09/2023 13:59:58,Living Room,98

09/09/2023 12:46:42,Main Bathroom,99

09/09/2023 10:20:52,Main Bedroom,100

09/08/2023 20:14:52,Dining Room,77

09/08/2023 19:12:16,Kitchen,81

09/06/2023 20:19:39,Dining Room,78

09/06/2023 19:23:09,Kitchen,83

09/04/2023 20:29:12,Dining Room,76

09/04/2023 19:31:31,Kitchen,88

09/02/2023 13:06:45,Living Room,93

09/02/2023 10:45:21,Main Bathroom,90

09/02/2023 08:18:39,Main Bedroom,95

09/01/2023 20:00:00,Dining Room,84

09/01/2023 19:00:00,Kitchen,90

To start the program, run CleaningManager.java

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Results** | **Actual Results** |
| **Test #1**  **testID: testSetupCleaningManager**  **Strategy:**  **Equivalence class - Loading existing and non-existing files** | **Preconditions:**   * **CleaningManager has been loaded successfully** * **The files rooms1.txt and logs1.txt exist as specified** * **The files rooms2.txt and logs2.txt exist as specified** * **The files rooms0.txt and logs0.txt do NOT exist in the file system**   **Steps:**   1. **Browse to specify rooms0.txt as the input file containing room information** 2. **Browse to specify logs0.txt as the input file containing logs of cleaning events** 3. ***See Expected Results*** 4. **Close the program** 5. **Start the program up again** 6. **Browse to specify rooms2.txt as the input file containing room information** 7. **Browse to specify logs2.txt as the input file containing logs of cleaning events** 8. ***See Expected Results*** 9. **Close the program** 10. **Start the program up again** 11. **Browse to specify rooms1.txt as the input file containing room information** 12. **Browse to specify logs1.txt as the input file containing logs of cleaning events** 13. ***See Expected Results*** | 3: If the user tries to view rooms0.txt and logs0.txt, then the program should prompt the user to specify a new file.  8: If the user tries to view rooms2.txt and logs2.txt, then the program successfully loads the file information and displays the main page of program options  13: If the user tries to view rooms1.txt and logs1.txt, then the program successfully loads the file information and displays the main page of program options | 3: The system outputs “One or both of the file names was incorrect, please re-enter the files.”  8: The program succesfully loads the files and displays options for output  13: The program succesfully loads the files and displays options for output |
| **Test #2**  **testID: testCleanRoomsFrequencyReport**  **Strategy:**  **Equivalence class - Testing valid inputs for room frequency**  **Exception - Testing -1 and 0 invalid inputs for room frequency**  **Boundary value - Testing lower(0, 1) and upper(6, 7) bounds for room frequency** | **Preconditions:**   * **CleaningManager has been loaded successfully using rooms1.txt and logs1.txt as input(as in Test #1)**   **Steps:**   1. **Select the option to view a report of the most frequently cleaned rooms** 2. **Enter 2 as the number of rooms to be in the report** 3. ***See Expected Results*** 4. **Enter 0 as the number of rooms to be in the report** 5. ***See Expected Results*** 6. **Enter -1 as the number of rooms to be in the report** 7. ***See Expected Results*** 8. **Enter 6 as the number of rooms to be in the report** 9. ***See Expected Results*** 10. **Enter 7 as the number of rooms to be in the report** 11. ***See Expected Reports*** 12. **Close the program** 13. **Start the program again** 14. **Browse to specify rooms2.txt as the input file containing room information** 15. **Browse to specify logs2.txt as the input file containing logs of cleaning events** 16. **Select the option to view a report of the most frequently cleaned rooms** 17. **Enter 1 as the number of rooms to be in the report** 18. ***See Expected Results*** 19. **Close the program** | 3: When the user inputs 2, information on cleaned room frequencies is presented to the user like the following:  Frequency of Cleanings [     Dining Room has been cleaned 6 times     Kitchen has been cleaned 6 times  ]  5: When the user inputs 0, the following error message should be presented to the user: “Number of rooms must be greater than 0.”  7: When the user inputs -1, the following error message should be presented to the user: “Number of rooms must be greater than 0.”  9: When the user inputs 6, information on cleaned room frequencies is presented to the user like the following:  Frequency of Cleanings [     Dining Room has been cleaned 6 times     Kitchen has been cleaned 6 times     Living Room has been cleaned 2 times     Main Bathroom has been cleaned 2 times     Main Bedroom has been cleaned 2 times     Entryway has been cleaned 0 times  ]  11: When the user inputs 7, information on cleaned room frequencies is presented to the user like the following:  Frequency of Cleanings [     Dining Room has been cleaned 6 times     Kitchen has been cleaned 6 times     Living Room has been cleaned 2 times     Main Bathroom has been cleaned 2 times     Main Bedroom has been cleaned 2 times     Entryway has been cleaned 0 times  ]  18: When the user inputs 1, the user should be presented with the following error message: “No rooms have been cleaned.” | 3: Output is the following: Frequency of Cleanings [  Dining Room has been cleaned 6 times  Kitchen has been cleaned 6 times  ]  5: Output is the following: Number of rooms must be greater than 0.  7: Output is the following: Number of rooms must be greater than 0.  9: Output is the following: Frequency of Cleanings [  Dining Room has been cleaned 6 times  Kitchen has been cleaned 6 times  Living Room has been cleaned 2 times  Main Bathroom has been cleaned 2 times  Main Bedroom has been cleaned 2 times  Entryway has been cleaned 0 times  ]  11: Output is the following: Frequency of Cleanings [  Dining Room has been cleaned 6 times  Kitchen has been cleaned 6 times  Living Room has been cleaned 2 times  Main Bathroom has been cleaned 2 times  Main Bedroom has been cleaned 2 times  Entryway has been cleaned 0 times  ]  18: Output is the following: No rooms have been cleaned. |
| **Test #3**  **testID: testRoomDateTimesReport**  **Strategy:**  **Equivalence class - Testing valid and invalid files for dates and times of when a room has been cleaned**  **Boundary value - Testing a room with no cleanings and an input file with no rooms or cleanings** | **Preconditions:**   * **CleaningManager has been loaded successfully using rooms1.txt and logs1.txt as input(as in Test #1)**   **Steps:**   1. **Select the option to view a report of when rooms were cleaned.** 2. ***See Expected Results*** 3. **Close Program** 4. **Start the program again** 5. **Browse to specify rooms2.txt as the input file containing room information** 6. **Browse to specify logs2.txt as the input file containing logs of cleaning events** 7. **Select the option to view a report of when rooms were cleaned.** 8. ***See Expected Results*** 9. **Close the program** | 2: When the user checks for a report of rooms1.txt and logs1.txt, the program should present to the user the following report:  Room Report [     Dining Room was cleaned on [        09/13/2023 20:08:05        09/11/2023 20:21:19        09/08/2023 20:14:52        09/06/2023 20:14:52        09/04/2023 20:29:12        09/01/2023 20:00:00     ]     Entryway was cleaned on [        (never cleaned)     ]     Kitchen was cleaned on [        09/13/2023 19:02:55        09/11/2023 19:09:10        09/08/2023 19:12:16        09/06/2023 19:23:09        09/04/2023 19:31:31        09/01/2023 19:00:00     ]     Living Room was cleaned on [        09/09/2023 13:59:58        09/02/2023 13:06:45     ]     Main Bathroom was cleaned on [        09/09/2023 12:46:42        09/02/2023 10:45:21     ]     Main Bedroom was cleaned on [        09/09/2023 10:20:52        09/02/2023 08:18:39     ]  ]  8: When the user checks for a report for rooms2.txt and log2.txt, the program should prompt the user with the following message: “No rooms have been cleaned.” | 2: Output is the following: Room Report [  Dining Room was cleaned on [  09/13/2023 20:08:05  09/11/2023 20:21:19  09/08/2023 20:14:52  09/06/2023 20:19:39  09/04/2023 20:29:12  09/01/2023 20:00:00  ]  Entryway was cleaned on [  (never cleaned)  ]  Kitchen was cleaned on [  09/13/2023 19:02:55  09/11/2023 19:09:10  09/08/2023 19:12:16  09/06/2023 19:23:09  09/04/2023 19:31:31  09/01/2023 19:00:00  ]  Living Room was cleaned on [  09/09/2023 13:59:58  09/02/2023 13:06:45  ]  Main Bathroom was cleaned on [  09/09/2023 12:46:42  09/02/2023 10:45:21  ]  Main Bedroom was cleaned on [  09/09/2023 10:20:52  09/02/2023 08:18:39  ]  ]  8: Output is the following: No rooms have been cleaned. |
| **Test #4**  **testID: testEstimatedRemainingLifeReport**  **Strategy:**  **Equivalence class - Testing valid inputs for checking the report of the vacuum bag remaining life**  **Exception - Testing dates and times with an invalid input format**  **Boundary value- Testing a date and time which puts the square feet over 5280 since last replacement** | **Preconditions:**   * **CleaningManager has been loaded successfully using rooms1.txt and logs1.txt as input(as in Test #1)**   **Steps:**   1. **Select the option to view a report of the estimated remaining vacuum bag life** 2. **Enter “09/02/2023 00:00:00”** 3. ***See Expected Results*** 4. **Enter “09/11/2023 02:10:27”** 5. ***See Expected Results*** 6. **Enter “09/03-2023 12:12:12”** 7. ***See Expected Results*** 8. **Enter “09/04/2023 02:60:03”** 9. ***See Expected Results*** 10. **Enter “01:02:03 04/05/2023”** 11. ***See Expected Results*** | 3: When the user inputs “09/02/2023 00:00:00”, the program should present the user with the following:  Vacuum Bag Report(last replaced 09/02/2023 00:00:00) [     Bag is overdue for replacement!  ]  5: When the user inputs “09/11/2023 02:10:27”, the program should present the user with the following:  Vacuum Bag Report(last replaced 09/11/2023 02:10:27) [     Bag is due for replacement in 4954 SQ FT  ]  7: When the user inputs “09/03-2023 12:12:12”, the user should be presented with the following error message: “Date & time must be in the format: MM/DD/YYYY HH:MM:SS”  9: When the user inputs “09/04/2023 02:60:03”, the user should be presented with the following error message: “Date & time must be in the format: MM/DD/YYYY HH:MM:SS”  11: When the user inputs “01:02:03 04/05/2023”, the user should be presented with the following error message: “Date & time must be in the format: MM/DD/YYYY HH:MM:SS” | 3: Output is the following: Vacuum Bag Report (last replaced 09/02/2023 00:00:00) [  Bag is overdue for replacement!  ]  5: Output is the following: Vacuum Bag Report (last replaced 09/11/2023 02:10:27) [  Bag is due for replacement in 4954 SQ FT  ]  7: Output is the following: Date & time must be in the format: MM/DD/YYYY HH:MM:SS  9: Output is the following: Date & time must be in the format: MM/DD/YYYY HH:MM:SS  11: Output is the following: Date & time must be in the format: MM/DD/YYYY HH:MM:SS |
| **Test #5**  **testID: testExitCleaningManager**  **Strategy:**  **Equivalence class - Testing that the cleaning manager closes successfully and without editing input files** | **Preconditions:**   * **CleaningManager has been loaded successfully using rooms1.txt and logs1.txt as input(as in Test #1)**   **Steps:**   1. **Select the option to close the software** 2. ***See Expected Results*** | 2: After the program closes, open the input files to make sure no changes were made to them as a result of using the software. | 2: The program exits successfully with no changes made to the input files. |