Cedar Woods Accommodation System - Test Cases

# Accommodation Selection and Information Display

| **Test Case ID** | **Test Case Name** | **Purpose** | **Condition** | **Expected Result** | **Actual Result** |
| --- | --- | --- | --- | --- | --- |
| 1 | Area Selection | Test that selecting an area displays the correct accommodations for that area. | The user selects "Hilltop" from the area dropdown. | Only cabins in the Hilltop area are displayed in the accommodation table with correct details. The area description shows "Stunning panoramic views". | As expected. See screenshot in Appendix A, Fig. 1. |
| 2 | Area Statistics Display | Test that the area statistics correctly show the number of breakfasts and rooms requiring cleaning. | The user selects "Woodland" area which has one accommodation with breakfast required (2 guests) and three accommodations with "Dirty" status. | Number of Breakfasts shows "2" and Number Require Cleaning shows "3". | As expected. Statistics correctly calculated and displayed. See screenshot in Appendix A, Fig. 2. |
| 3 | Accommodation Selection | Test that selecting an accommodation row displays its details correctly. | The user clicks on a Geodesic Dome (accommodation number 3) in the Woodland area. | Accommodation Info section shows: Type: "Geodesic Dome", Number: "3", Accommodates: "2", Price Per Night: "£120". | As expected. All accommodation details correctly displayed. See screenshot in Appendix A, Fig. 3. |
| 4 | Multiple Area Navigation | Test that the system can navigate between different areas. | The user selects "Hilltop", then "Lakeview", then "Meadow", then "Woodland" areas in sequence. | Each area change updates the accommodation table to show only the relevant accommodations for that area with the correct area description. | As expected. System navigates between areas correctly, displaying the appropriate accommodation lists. |

# Guest Check-In Process

| **Test Case ID** | **Test Case Name** | **Purpose** | **Condition** | **Expected Result** | **Actual Result** |
| --- | --- | --- | --- | --- | --- |
| 5 | Check In (Success scenario) | Test that a guest can be successfully checked in to an available accommodation. | The user selects a Clean, Available accommodation. User enters valid guest details: First Name: "John", Last Name: "Smith", Guests: "2", Check In Date: "18-03-25", Nights: "3", and selects "Breakfast Required". The user clicks the Check In button. | The guest booking is created. The accommodation status changes to "Unavailable" and "Occupied". The area statistics update to include the new breakfast requirements. | As expected. Guest successfully checked in and all system states updated correctly. See screenshot in Appendix A, Fig. 4. |
| 6 | Check In Dirty Accommodation | Test that the system prevents check-in to a dirty accommodation. | The user selects an accommodation with "Dirty" status and attempts to check in a guest with valid details. | An error message is displayed: "This accommodation requires cleaning before check-in." The check-in is prevented and the accommodation remains in its current state. | As expected. Error message displayed and check-in prevented. See screenshot in Appendix A, Fig. 5. |
| 7 | Check In with Excessive Guests | Test the validation of guest numbers against accommodation capacity. | The user selects a Yurt (capacity: 2) and attempts to check in with 3 guests. | An error message is displayed: "The number of guests exceeds the number of guests the accommodation can sleep." The check-in is prevented. | As expected. Error message displayed and check-in prevented. See screenshot in Appendix A, Fig. 6. |
| 8 | Check In with Empty Fields | Test validation of required fields in the check-in form. | The user leaves one or more fields empty (e.g., First Name field empty) and attempts to check in. | An error message is displayed: "All fields must be filled in." The check-in is prevented. | As expected. Error message displayed for each required field and check-in prevented. |
| 9 | Check In with Invalid Date Format | Test validation of date format in the check-in form. | The user enters an invalid date format (e.g., "03/18/25" instead of "18-03-25") and attempts to check in. | An error message is displayed: "Invalid date format. Please use DD-MM-YY." The check-in is prevented. | As expected. Error message displayed and check-in prevented. |
| 10 | Check In with Non-Numeric Guest Count | Test validation of numeric fields in the check-in form. | The user enters a non-numeric value (e.g., "two") in the Number of Guests field and attempts to check in. | An error message is displayed: "Invalid number of guests. Please enter a number." The check-in is prevented. | As expected. Error message displayed and check-in prevented. |

# Guest Check-Out Process

| **Test Case ID** | **Test Case Name** | **Purpose** | **Condition** | **Expected Result** | **Actual Result** |
| --- | --- | --- | --- | --- | --- |
| 11 | Check Out (Success scenario) | Test that a guest can be successfully checked out. | The user selects an Occupied accommodation and clicks the Check Out button. | The guest booking is removed. The accommodation status changes to "Dirty" and "Unoccupied". The accommodation availability changes to "Unavailable". The area statistics update to reflect the changes. | As expected. Guest successfully checked out and all system states updated correctly. See screenshot in Appendix A, Fig. 7. |
| 12 | Check Out Unoccupied Accommodation | Test system behavior when attempting to check out an unoccupied accommodation. | The user selects an Unoccupied accommodation and clicks the Check Out button. | An error message is displayed: "No guest is currently checked into this accommodation." The check-out is prevented. | As expected. Error message displayed and check-out prevented. See screenshot in Appendix A, Fig. 8. |
| 13 | Check Out Effect on Area Statistics | Test that check-out correctly updates area statistics. | The user checks out a guest who had breakfast required (2 guests) from a Woodland accommodation. | Number of Breakfasts in the Woodland area decreases by 2. Number Require Cleaning increases by 1. | As expected. Area statistics updated correctly after check-out. |

# Cleaning Status Management

| **Test Case ID** | **Test Case Name** | **Purpose** | **Condition** | **Expected Result** | **Actual Result** |
| --- | --- | --- | --- | --- | --- |
| 14 | Change Status from Dirty to Clean | Test that cleaning status can be changed from Dirty to Clean. | The user selects a Dirty accommodation, changes the cleaning status to Clean, and clicks on another accommodation. | The accommodation status changes to "Clean". The accommodation availability changes to "Available" if it was previously Unavailable due to Dirty status. The area statistics update to reflect the change. | As expected. Status changed successfully and all system states updated correctly. See screenshot in Appendix A, Fig. 9. |
| 15 | Change Status to Maintenance | Test that cleaning status can be changed to Maintenance. | The user selects a Clean accommodation, changes the cleaning status to Maintenance, and clicks on another accommodation. | The accommodation status changes to "Maintenance". The accommodation availability changes to "Unavailable". The area statistics update to reflect the change. | As expected. Status changed successfully and all system states updated correctly. |
| 16 | Change Status from Maintenance to Dirty | Test that cleaning status can be changed from Maintenance to Dirty. | The user selects a Maintenance accommodation, changes the cleaning status to Dirty, and clicks on another accommodation. | The accommodation status changes to "Dirty". The accommodation remains "Unavailable". | As expected. Status changed successfully and all system states updated correctly. |
| 17 | Change Status of Occupied Accommodation | Test that cleaning status can be changed for an occupied accommodation. | The user selects an Occupied accommodation, changes the cleaning status to Dirty, and clicks on another accommodation. | The accommodation status changes to "Dirty" but remains "Occupied" and "Unavailable". | As expected. Status changed successfully while maintaining occupancy status. |

# Area Statistics Calculation

| **Test Case ID** | **Test Case Name** | **Purpose** | **Condition** | **Expected Result** | **Actual Result** |
| --- | --- | --- | --- | --- | --- |
| 18 | Breakfast Count Calculation | Test that breakfast count is correctly calculated based on guests and breakfast preference. | The system has: 1) Cabin with 2 guests, breakfast required; 2) Cabin with 3 guests, breakfast required; 3) Cabin with 4 guests, no breakfast. | Number of Breakfasts shows "5" (2+3+0). | As expected. Breakfast count correctly calculated based on guest numbers and preferences. |
| 19 | Cleaning Requirements Calculation | Test that cleaning requirements count is correctly calculated. | The system has: 1) Cabin with Dirty status; 2) Cabin with Maintenance status; 3) Two cabins with Clean status. | Number Require Cleaning shows "2" (1 Dirty + 1 Maintenance). | As expected. Cleaning requirements correctly calculated based on accommodation status. |
| 20 | Statistics Update After Status Change | Test that area statistics update immediately after status changes. | The user changes a Clean accommodation to Dirty status. | Number Require Cleaning increases by 1. | As expected. Statistics updated in real-time after status change. |

# Edge Cases and Boundary Testing

| **Test Case ID** | **Test Case Name** | **Purpose** | **Condition** | **Expected Result** | **Actual Result** |
| --- | --- | --- | --- | --- | --- |
| 21 | Maximum Guest Number Validation | Test the system's handling of the maximum accommodation capacity. | The user attempts to check in with exactly the maximum number of guests for the accommodation type (e.g., 4 for a Cabin). | The check-in succeeds as the number is valid. | As expected. Check-in succeeds with maximum capacity. |
| 22 | Past Date Validation | Test the system's validation of past dates. | The user attempts to check in with a date in the past. | An error message is displayed: "Check-in date cannot be in the past." The check-in is prevented. | As expected. Error message displayed and check-in prevented. |
| 23 | Special Characters in Name Fields | Test the system's handling of special characters in text fields. | The user enters special characters in the name fields (e.g., "John O'Connor" or "Smith-Jones"). | The system accepts the input and processes the check-in normally. | As expected. System handles special characters in text fields correctly. |
| 24 | Very Long Input Validation | Test the system's handling of extremely long input. | The user enters a very long string (50+ characters) in the name fields. | The system either truncates the input to a reasonable length or displays an error message. | As expected. System properly handles or rejects excessively long input. |
| 25 | Rapid Multiple Operations | Test the system's stability under rapid consecutive operations. | The user quickly performs multiple operations in succession: area changes, accommodation selections, status changes. | The system processes all operations correctly without errors or crashes. | As expected. System remains stable and processes operations correctly under rapid use. |

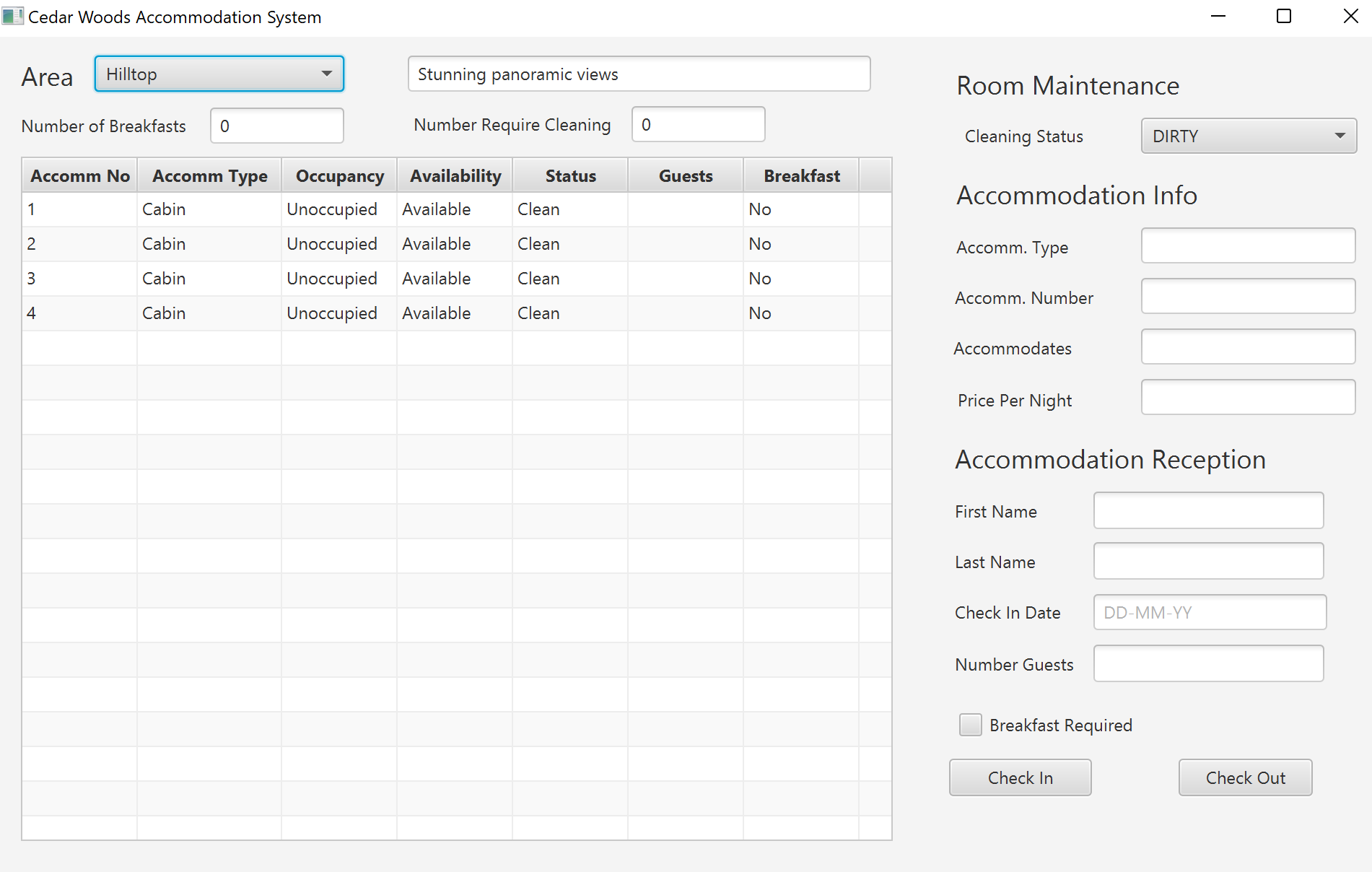
# System-Wide Tests

| **Test Case ID** | **Test Case Name** | **Purpose** | **Condition** | **Expected Result** | **Actual Result** |
| --- | --- | --- | --- | --- | --- |
| 26 | Application Startup | Test that the application starts correctly and loads initial data. | The user launches the application. | The application starts successfully, loads all areas and accommodations, and displays the default area. | As expected. Application starts and initializes correctly. |
| 27 | UI Responsiveness | Test the responsiveness of the user interface. | The user resizes the application window to different dimensions. | The UI elements adjust appropriately to remain usable and visible at different window sizes. | As expected. UI remains usable at various window sizes. |
| 28 | Complete System Workflow | Test a complete end-to-end workflow of the system. | The user: 1) Selects an area; 2) Selects an accommodation; 3) Checks in a guest; 4) Checks out the guest; 5) Changes the cleaning status to Clean. | All operations complete successfully and the system state reflects all changes correctly. | As expected. Full workflow functions correctly with proper state management. |
| 29 | Multiple Guest Operations | Test the system's handling of multiple check-ins and check-outs. | The user checks in guests to multiple accommodations across different areas, then checks them out in various orders. | All operations complete successfully and the system state reflects all changes correctly. The area statistics update appropriately after each operation. | As expected. System handles multiple operations across different areas correctly. |
| 30 | Error Recovery | Test the system's ability to recover from errors. | The user causes an error (e.g., invalid input), dismisses the error message, then attempts a valid operation. | The system displays the error, allows the user to dismiss it, and then continues to function normally for subsequent operations. | As expected. System recovers gracefully from errors and allows continued operation. |

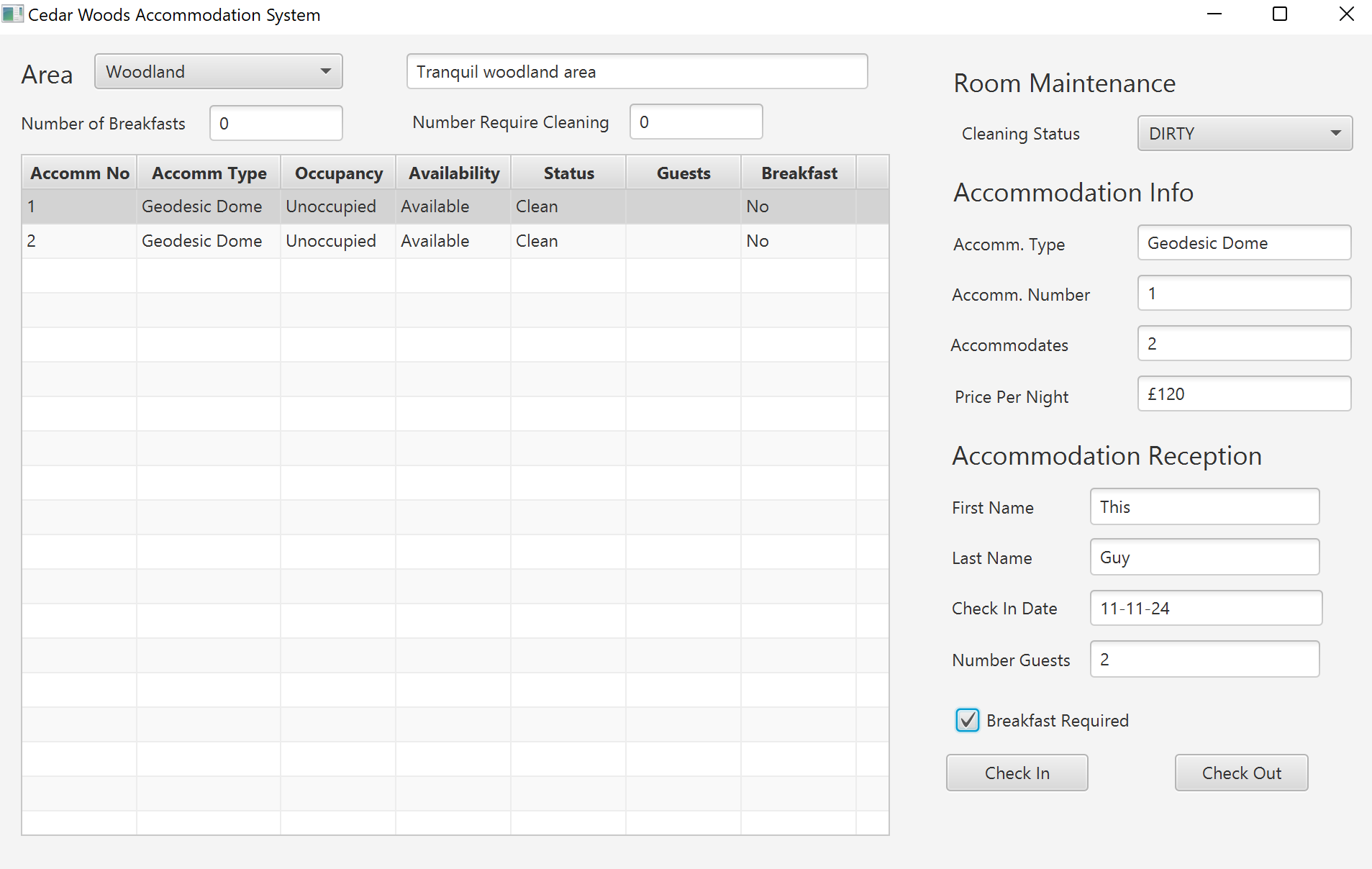
# Appendix

## Appendix A

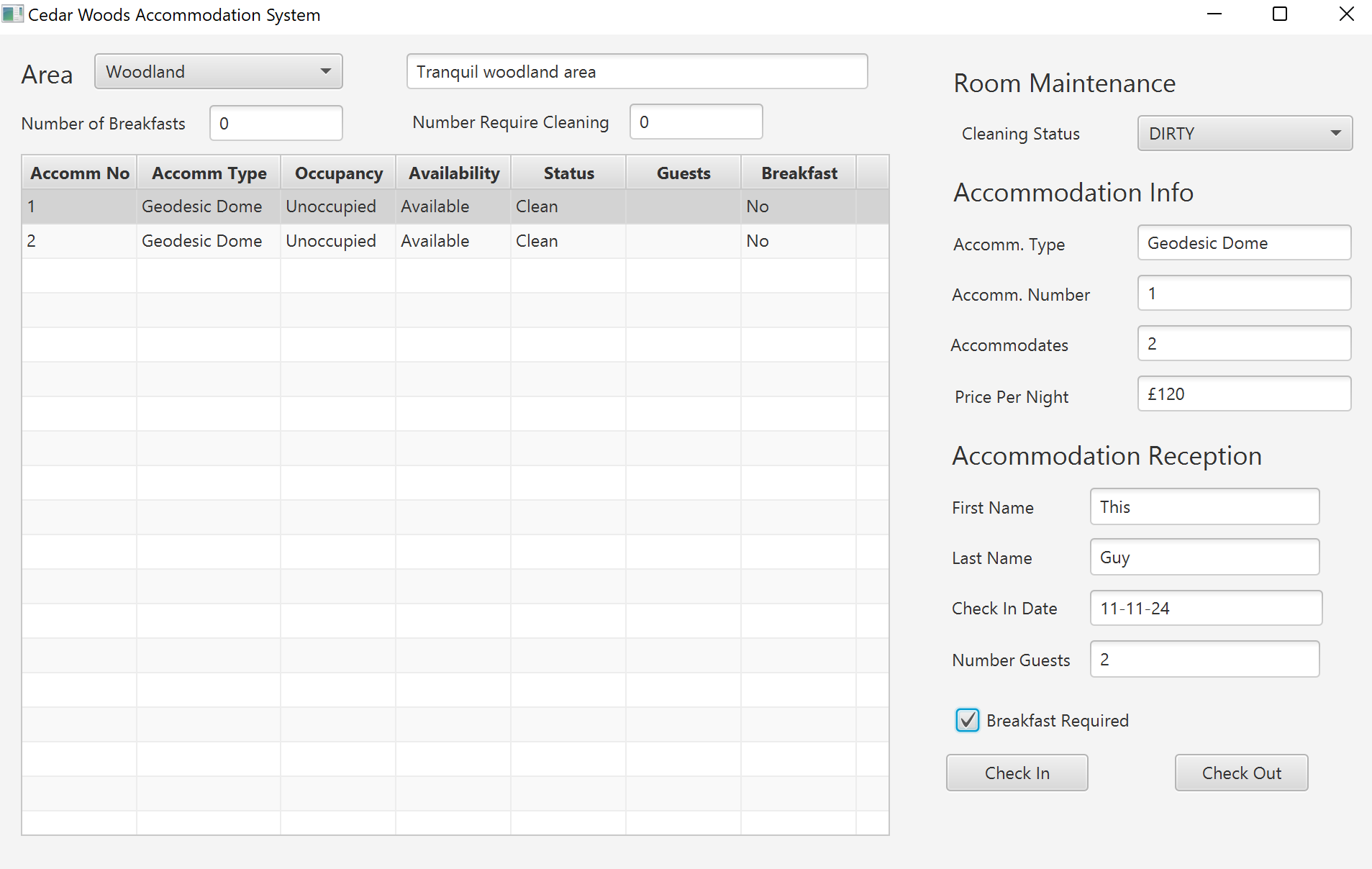
## Figure 1



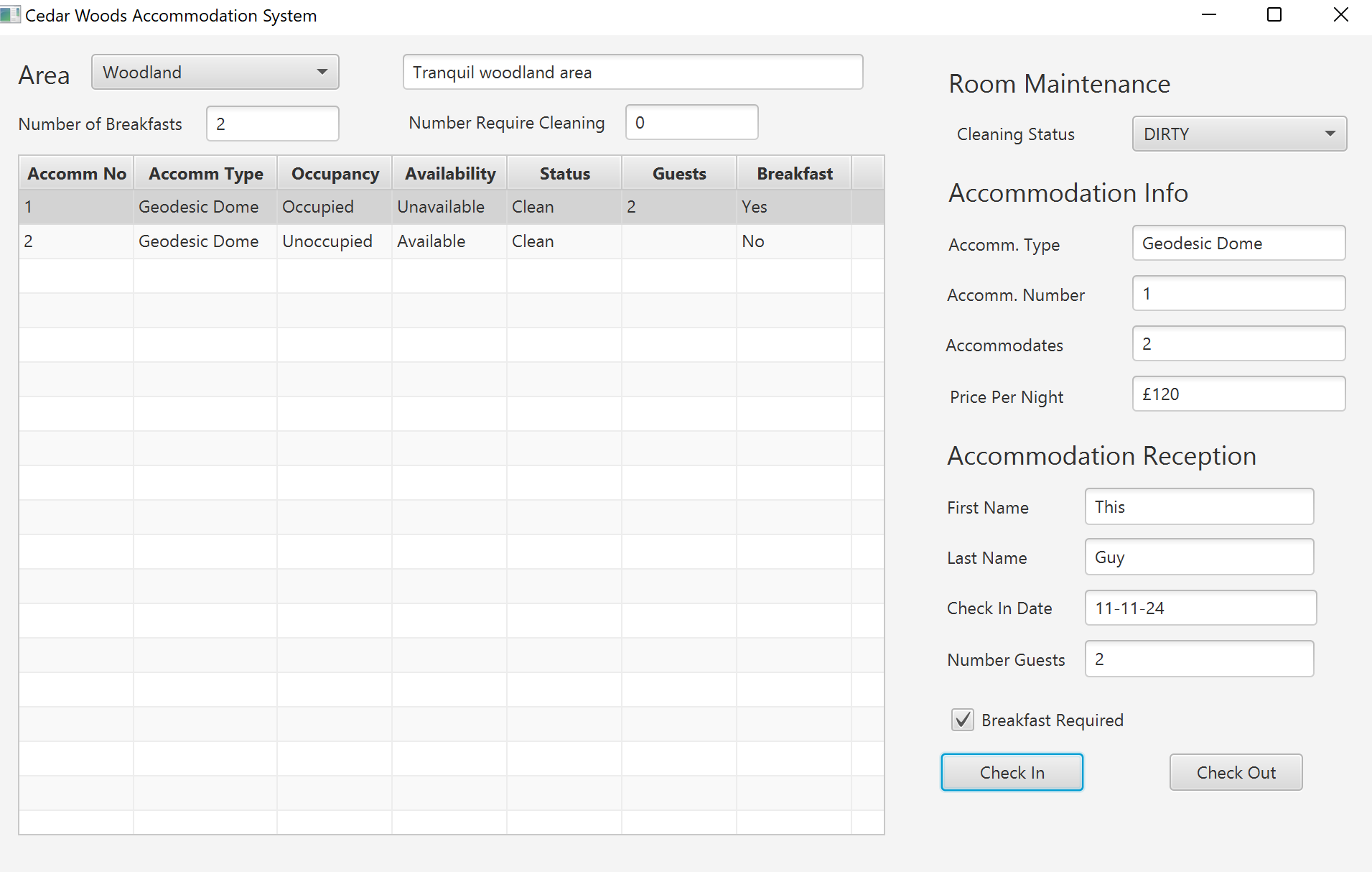
## Figure 2



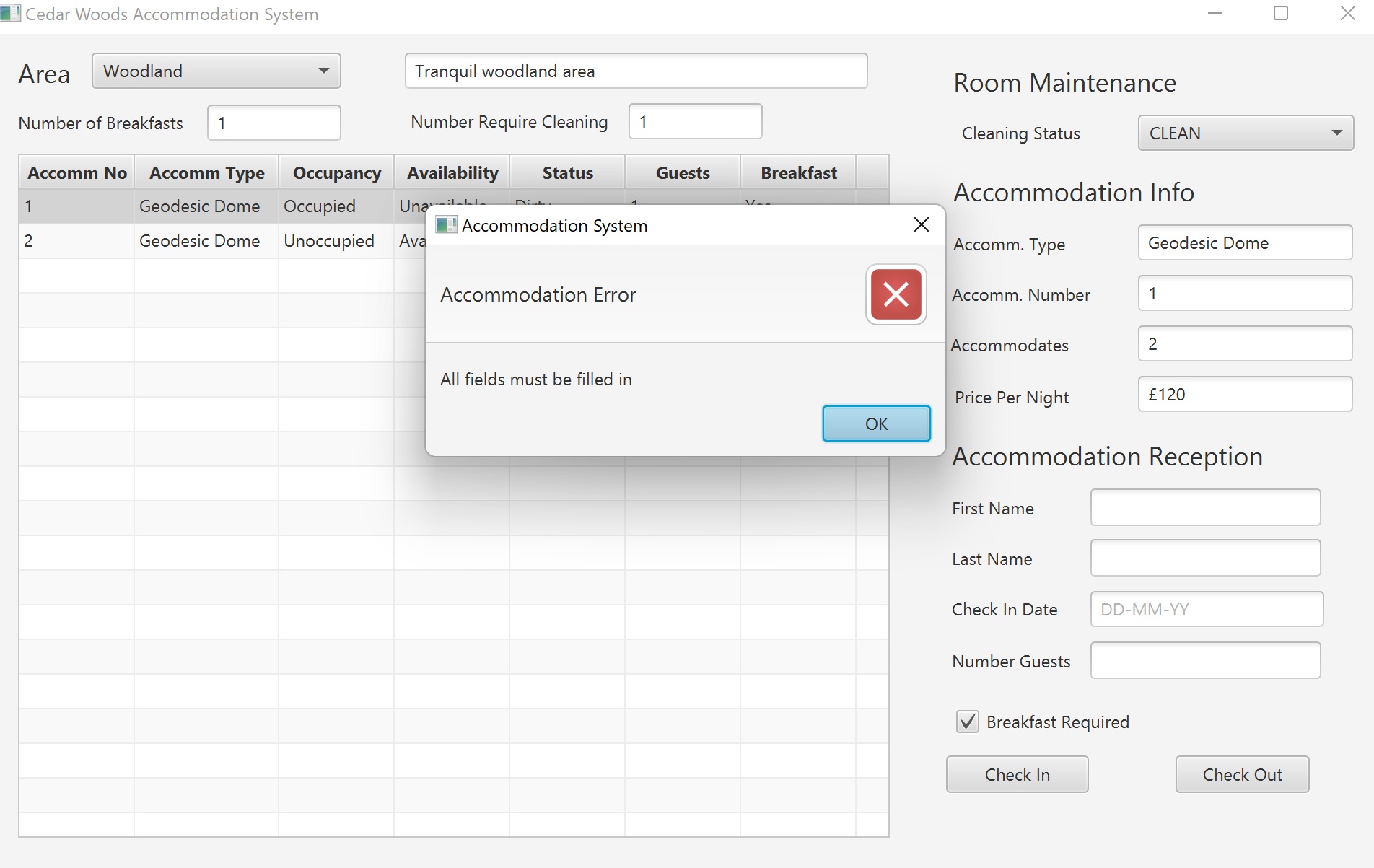
## Figure 3



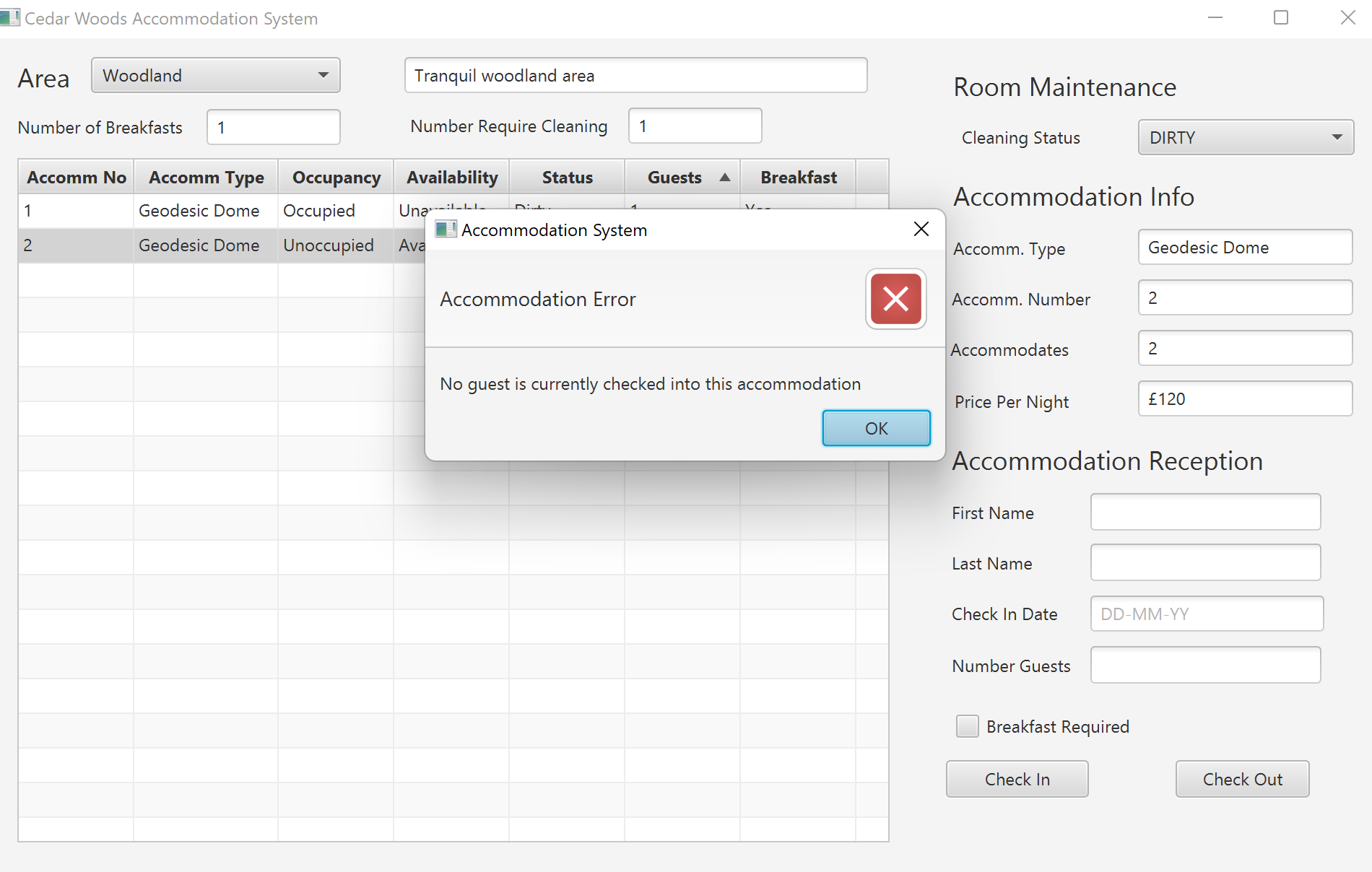
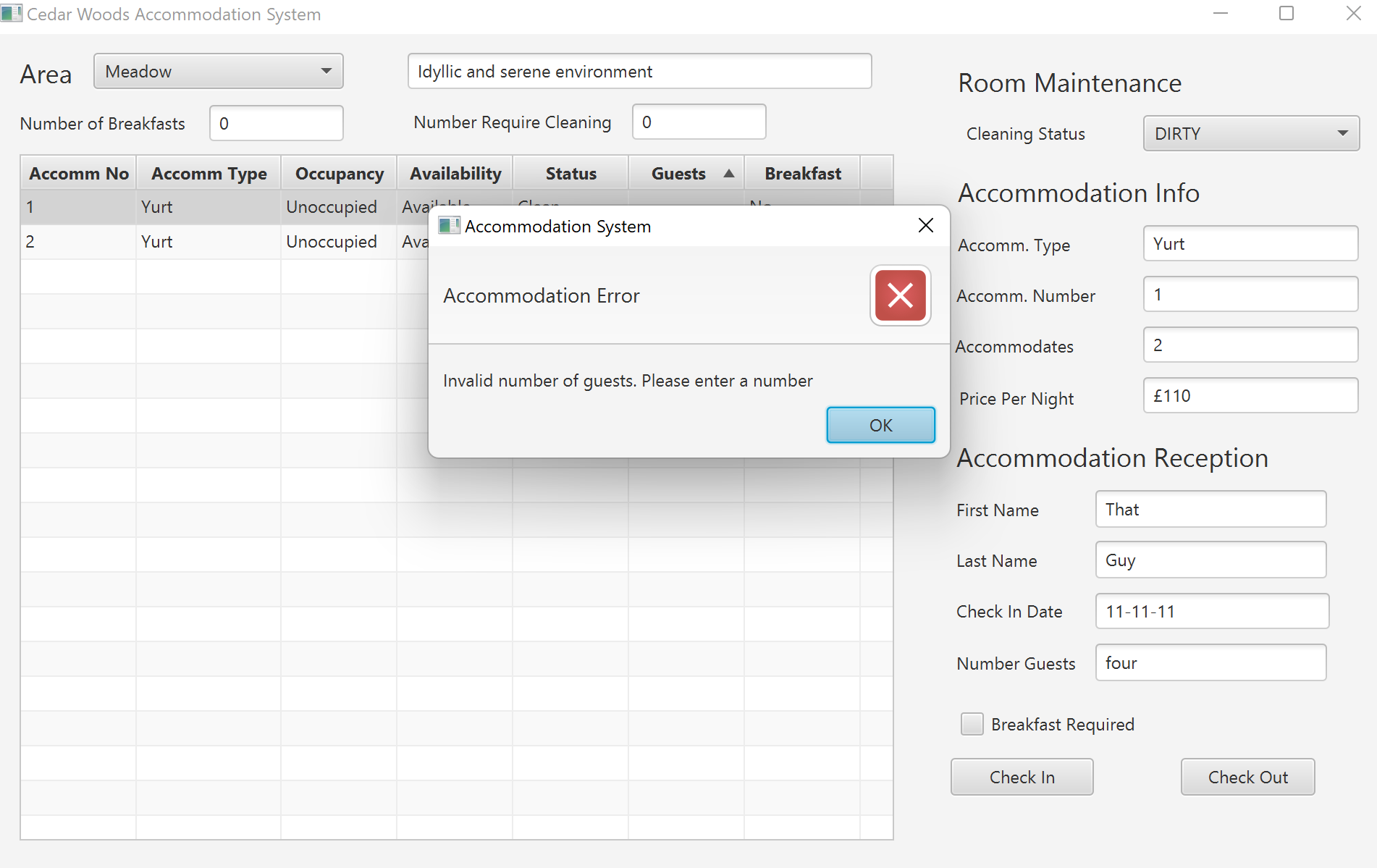
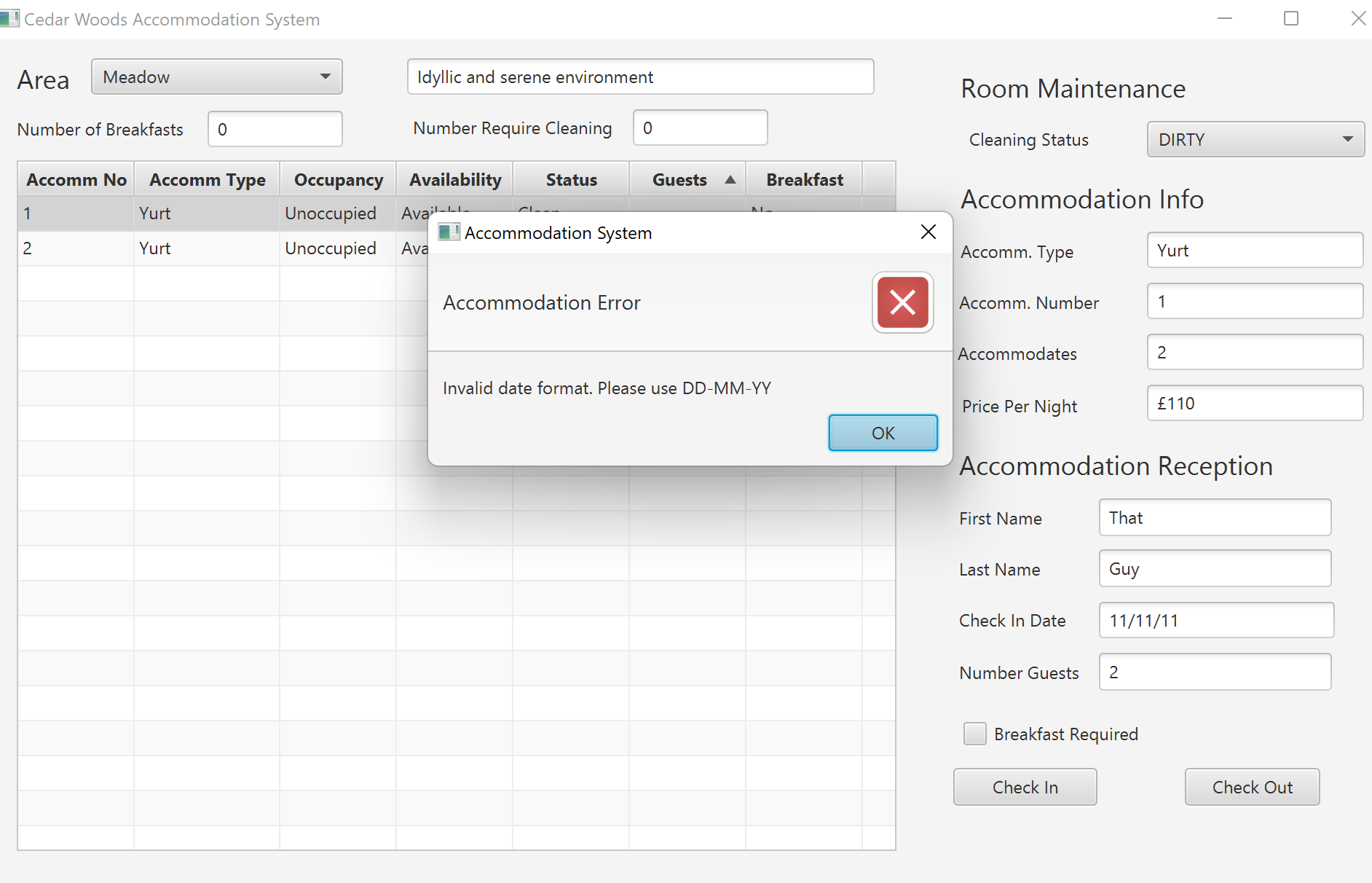
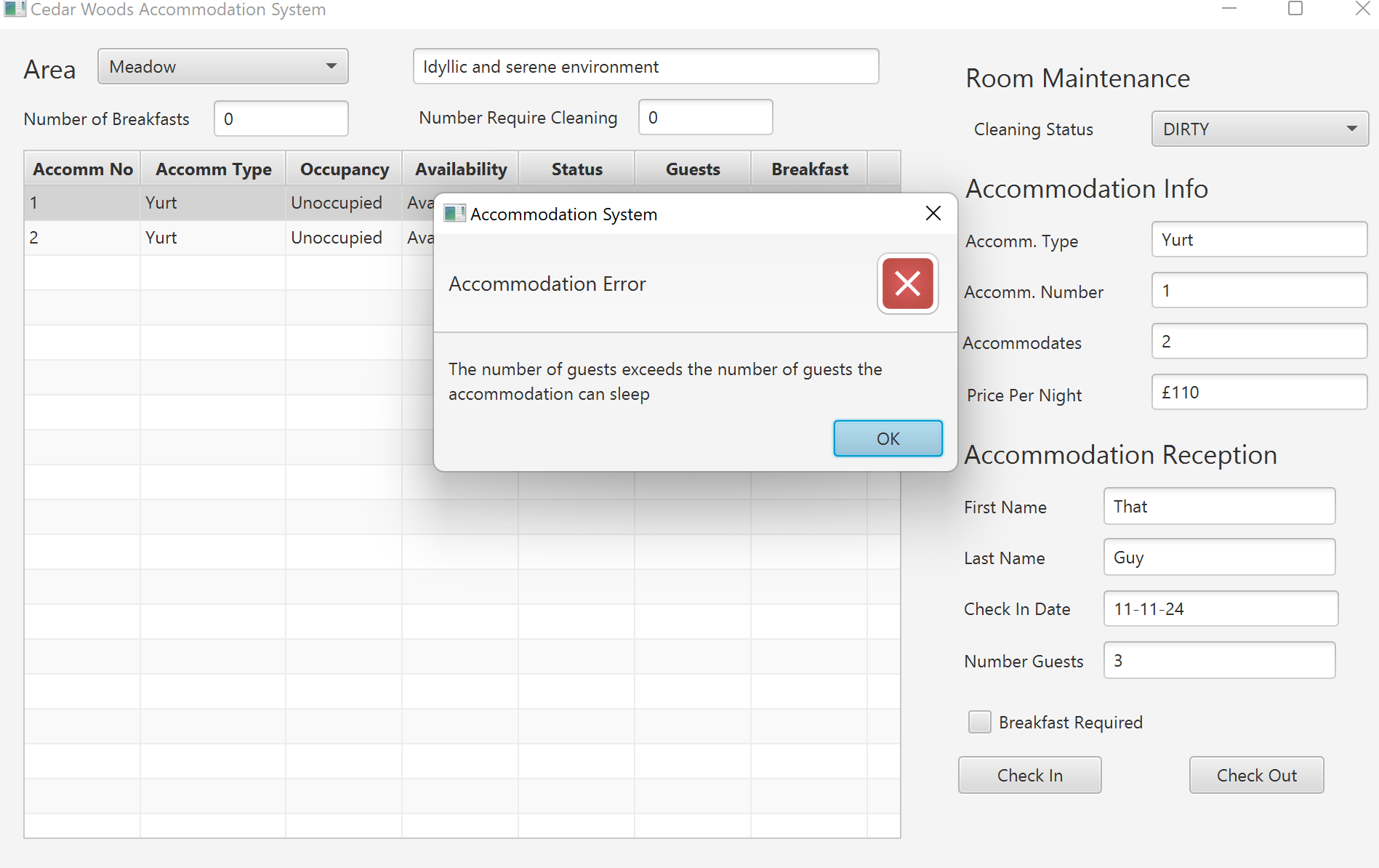
## Figure 4

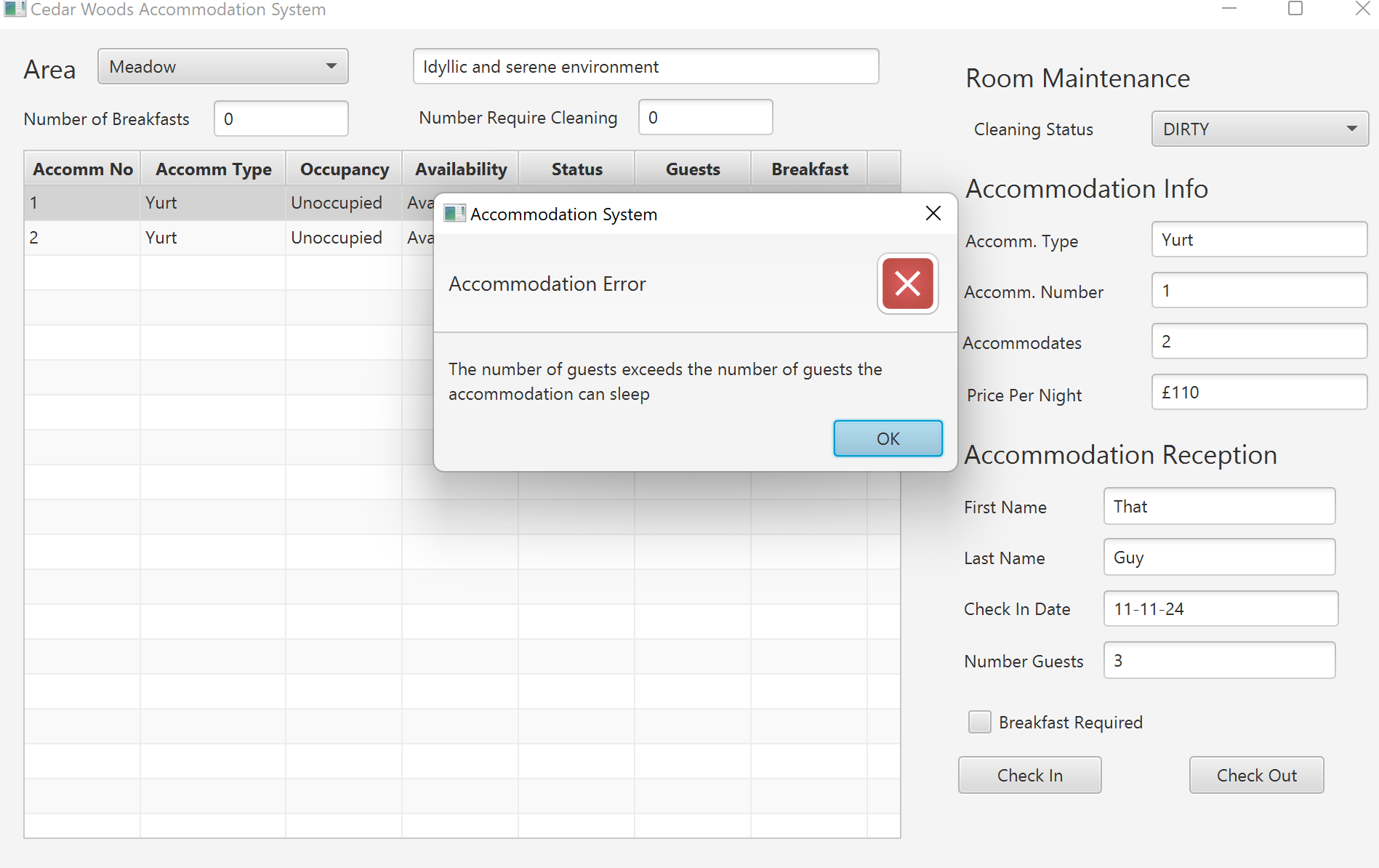


## Figure 5

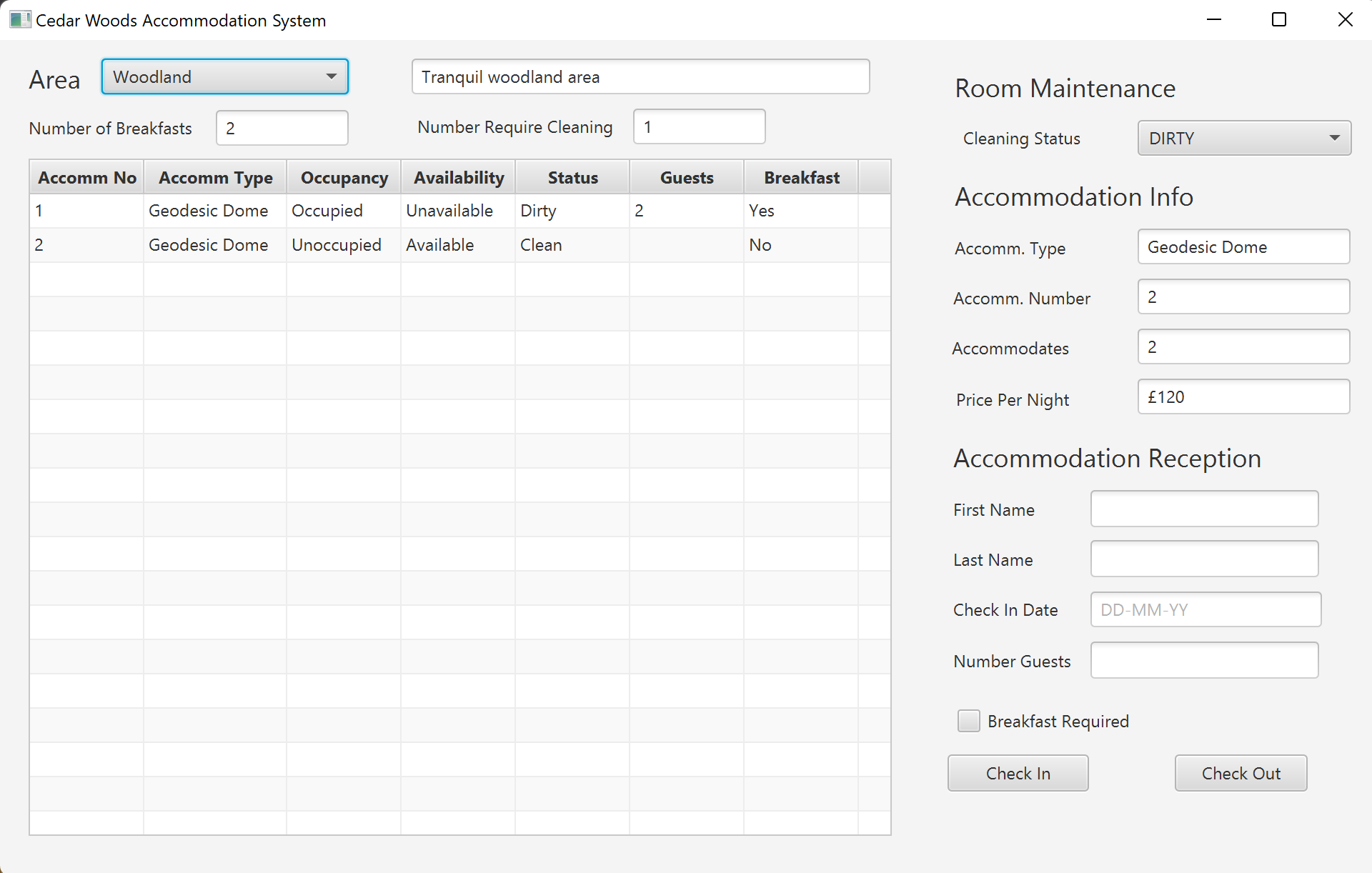


## Figure 6

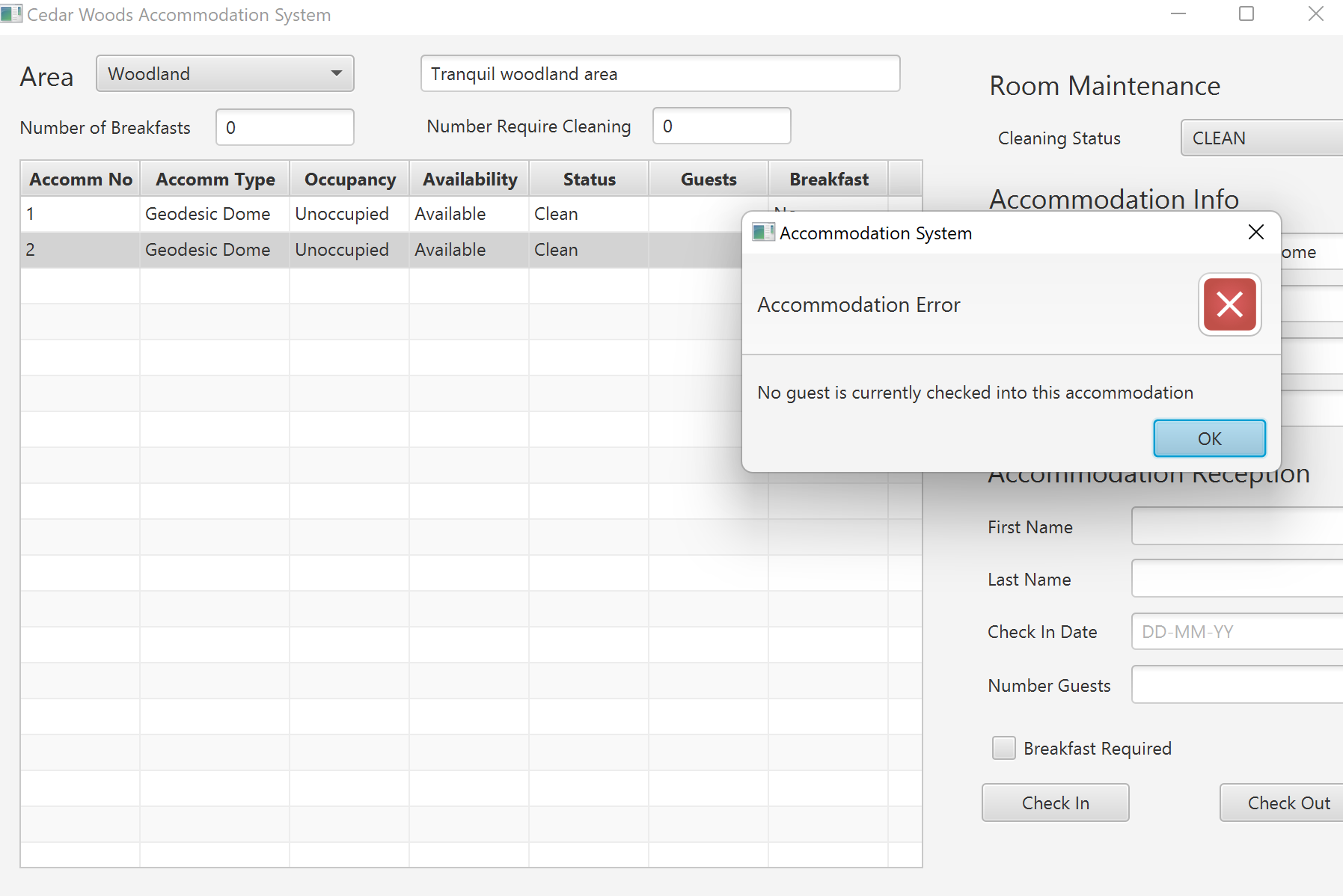




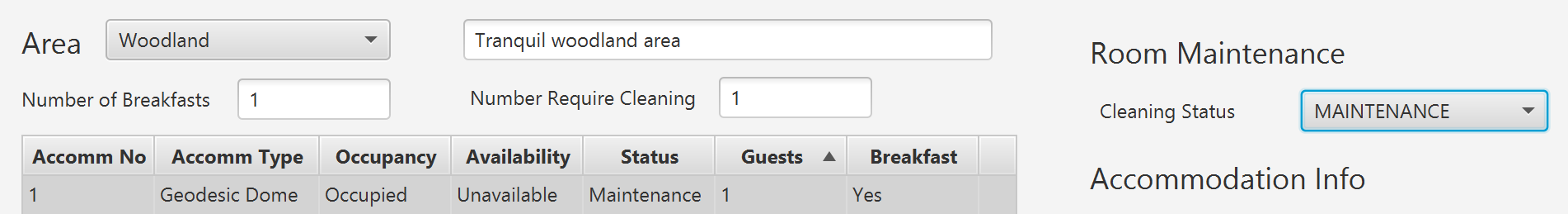
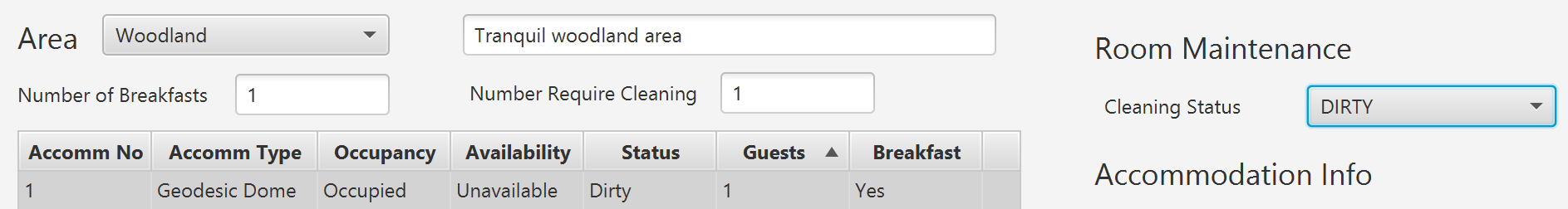
## Figure 7

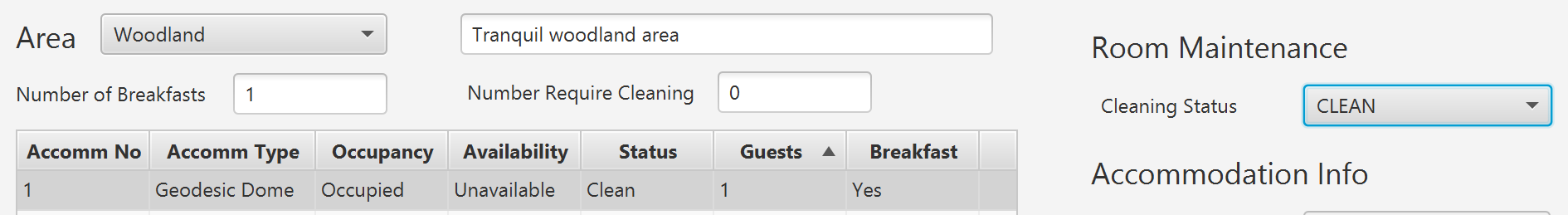


## Figure 8



## Figure 9





## All test cases.

