**Testing documentation -**

In this document we have produced tests in order to ensure that the finished program acts as desired and meets all of the conditions included in the project brief.

Failure to meet any of the following tests will result in the program being reviewed further to correct these issues. The corrections will then be reviewed and tested again in order to validate the system and ensure it is completely functional.

Each test has been included in the appropriate tables and includes the following –

Test number – this number represents the test and will correspond with the images in the bibliography. These images will act as proof of the test being carried out. For example, test 1 will be matched with ‘1-‘ in the bibliography.

Test case – the actions that will be performed during this test in order to produce the desired result

Purpose – the purpose of the test and the reason this test is being carried out

Expected result – this is what the system is expected to produce when carried out. This is the condition that will be compared with the actual result in order to understand if the test was successful or unsuccessful in producing the desired result.

Actual result – what the system actually produces. If this differs from the expected result then the test has failed and the code must be updated. If the actual result is the same as the expected result then the test was successful.

At the bottom of this document I have included the code used for Junit testing. This is the code used to test each of the classes and was ran during the programs development to print more appropriate error messages.

Hall Manager specific tests –

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test  Number | Test Case | Purpose | Expected result | Actual result |
| 1 | Make two lease agreements under the same lease number | Ensure the same lease number can’t be used more than once | The system will reject the input and tell them what the issue was | System provided the desired output |
| 2 | Make two lease agreements under the same student name | Ensure the same student can’t have multiple rooms | The system will reject the input and tell them what the issue was | System provided the desired output |
| 3 | Lease an offline room to a student | Ensure the system won’t allow anything other than a clean or dirty room to be provided to a student | The system will reject the input and tell them what the issue was | System provided the desired output |
| 4 | Add a toom with a lease number greater than 999,999 | Ensure the system won’t allow a number greater than 999,999 | The system will reject the input and tell them what the issue was | System provided the desired output |
| 5 | Lease a room to a student with the occupation status ‘unoccupied’ | Ensure the system takes valid input without issues | The system will accept the input and add it to the table | System provided the desired output |
| 6 | Lease a room to a student with the occupation status ‘occupied’ | Ensure the system takes valid input without issues | The system will accept the input and add it to the table | System provided the desired output |
| 7 | Lease a room to a student a lease number that has not been used yet | Ensure the system takes valid input without issues | The system will accept the input and add it to the table | System provided the desired output |
| 8 | Lease a room to a student that is not already assigned to a lease | Ensure the system takes valid input without issues | The system will accept the input and add it to the table | System provided the desired output |
| 9 | Lease a room to a student with a lease less than 48 month in length | Ensure the system takes valid input without issues | The system will accept the input and add it to the table | System provided the desired output |
| 10 | Delete a lease currently occupied with data | Ensure the system takes valid input without issues | The system will accept the input and remove it from the table | System provided the desired output |
| 11 | Delete a lease that holds no data | Ensure the system won’t allow redundant deletion | The system will reject the input and tell them what the issue was | System provided the desired output |

Warden specific tests –

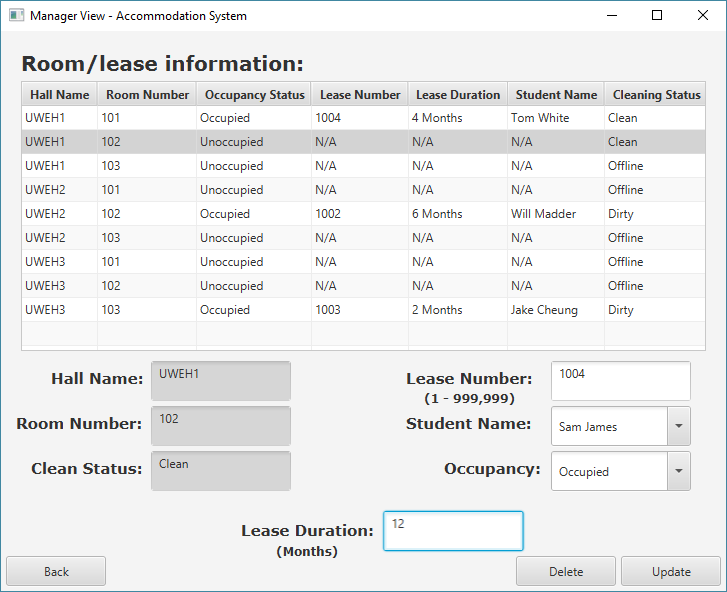
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test  Number | Test Case | Purpose | Expected result | Actual result |
| 12 | Change a dirty room to clean | Ensure the system takes valid input without issues | The system will accept the input and update the information in the table | System accepted the input and did as expected |
| 13 | Change a clean room to dirty | Ensure the system takes valid input without issues | The system will accept the input and update the information in the table | System accepted the input and did as expected |
| 14 | Change an offline room into a clean/dirty one | Ensure the system takes valid input without issues | The system will accept the input and update the information in the table | System accepted the input and did as expected |
| 15 | Change a clean/dirty (unoccupied) room into an offline room | Ensure the system takes valid input without issues | The system will accept the input and update the information in the table | System accepted the input and did as expected |
| 16 | Change a clean/dirty (occupied) room into an offline room | Ensure the system won’t take this invalid input | The system will reject this input and tell the user what the issue is | System rejected the input and did as expected |

Menu specific tests -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test  Number | Test Case | Purpose | Expected result | Actual result |
| 17 | Select the Hall manager option | Ensure the system takes valid input without issues | The system will accept the input and take the user to the Manager page, allowing for editing of the appropriate information | The system took the user to the appropriate page, as expected |
| 18 | Select the Warden page | Ensure the system takes the valid input without issues | The system will accept the input and take the user to the Warden page, allowing for editing of the appropriate information | The system took the user to the appropriate page, as expected |
| 19 | Select the All page | Ensure the system takes the valid input without issues | The system will accept the input and take the user to the All page, allowing for editing of the appropriate information | The system took the user to the appropriate page, as expected |

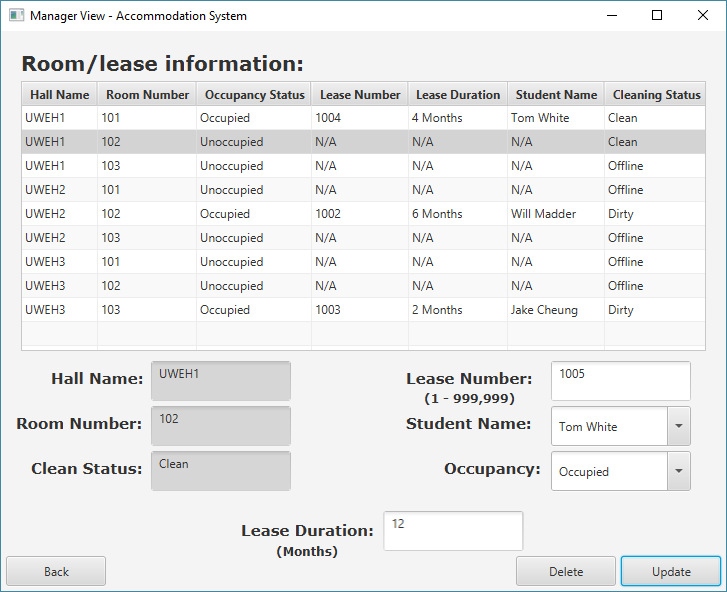
All page specific tests -

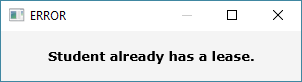
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test  Number | Test Case | Purpose | Expected result | Actual result |
| 20 | Add a new student | Ensure that a new student with appropriate information can be added to the system | The system will accept the student information and add it to the appropriate table | The system accepted the new student, as expected |
| 21 | Add a new room | Ensure that a new hall can be added with the appropriate information | The system will accept the new room and add it to the appropriate table | The system accepted the new table, as expected |
| 22 | Add a new hall | Ensure that a new hall can be added with the appropriate information | The system will accept the new hall and add it to the appropriate table | The system accepted the new table, as expected |
| 23 | Add a student missing a part of their name | Ensure the system won’t take this invalid input | The system will reject this input and inform the user as to what the issue is | The system failed to produce the appropriate result, it accepted only a first name |
| 24 | Add a student with an inappropriate ID | Ensure the system won’t take this invalid input | The system will reject this input and inform the user as to what the issue is | The system did not accept the input and displayed the appropriate message as desired. |
| 25 | Add a hall with an inappropriate ID | Ensure the system won’t take this invalid input | The system will reject this input and inform the user as to what the issue is | The system did not accept the input and displayed the appropriate message as desired. |
| 26 | Add a room with an inappropriate ID | Ensure the system won’t take this invalid input | The system will reject this input and inform the user as to what the issue is | The system did not accept the input and displayed the appropriate message as desired. |
| 27 | Add a room with the same ID as an existing room | Ensure the system won’t take this invalid input | The system will reject this input and inform the user as to what the issue is | The system did not accept the input and displayed the appropriate message as desired. |
| 28 | Add a student with the same ID as an existing student | Ensure the system won’t take this invalid input | The system will reject this input and inform the user as to what the issue is | The system did not accept the input and displayed the appropriate message as desired. |
| 29 | Add a hall with the same ID as an existing hall | Ensure the system won’t take this invalid input | The system will reject this input and inform the user as to what the issue is | The system did not accept the input and displayed the appropriate message as desired. |
| 23 fix | Add a student to the system with a missing name field | Ensure the system wont take this invalid input | The system will reject the input and inform the user what the issue is. | The system did not accept the input, as expected. The issue has been fixed. |

1-

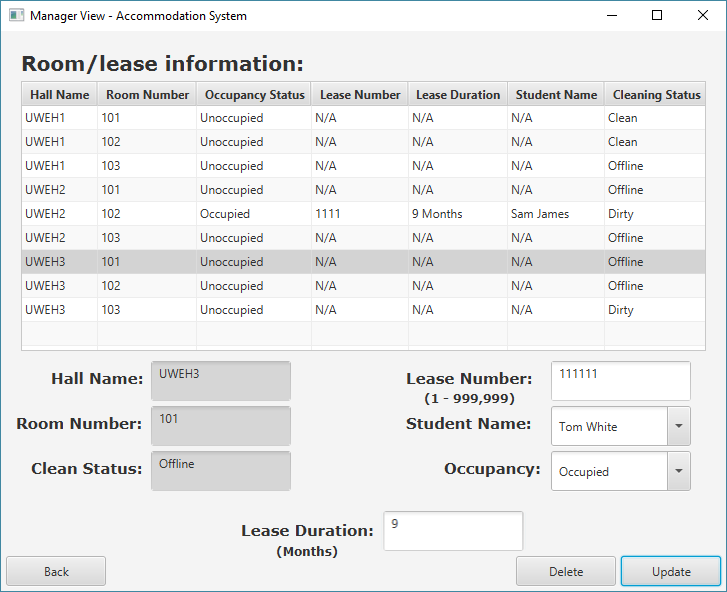


2 -

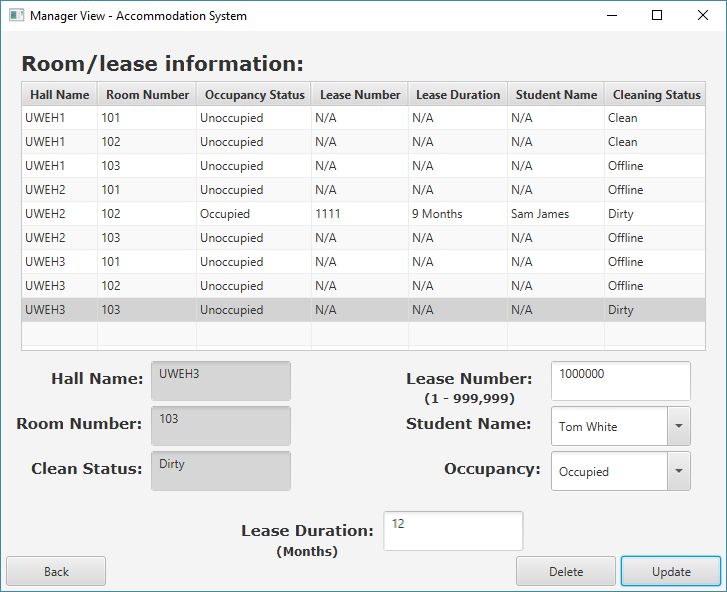




3 –

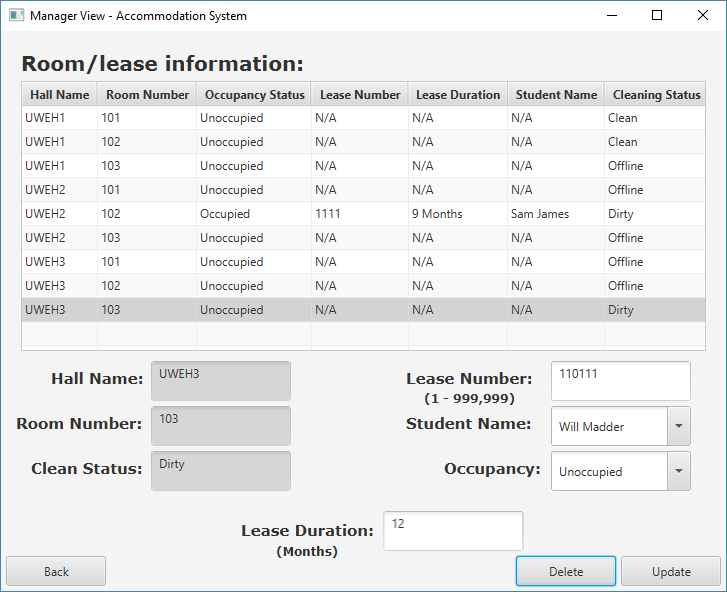


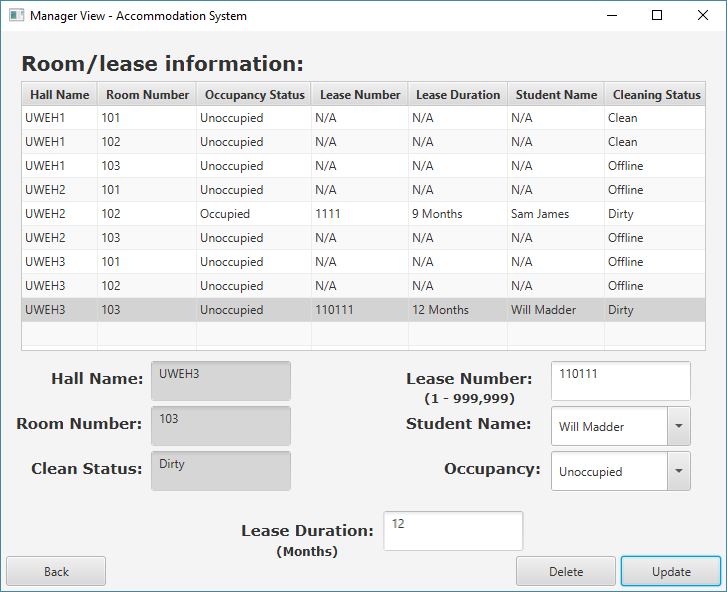


4 - 

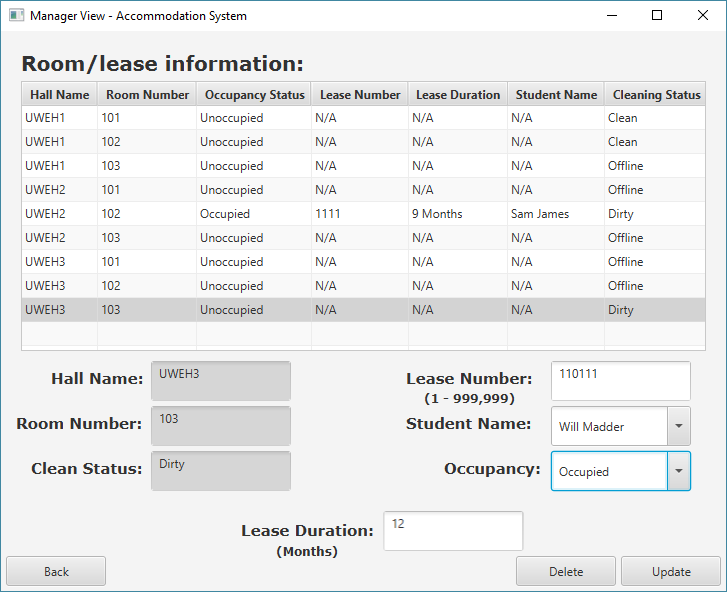


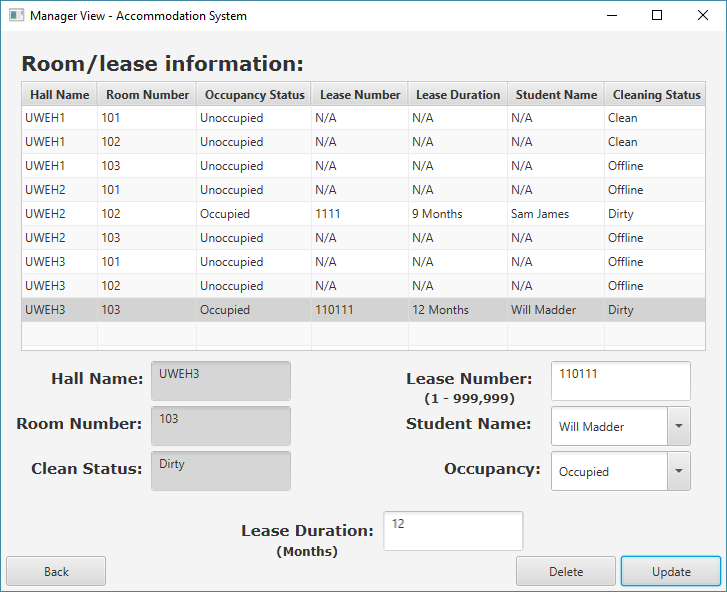
5-



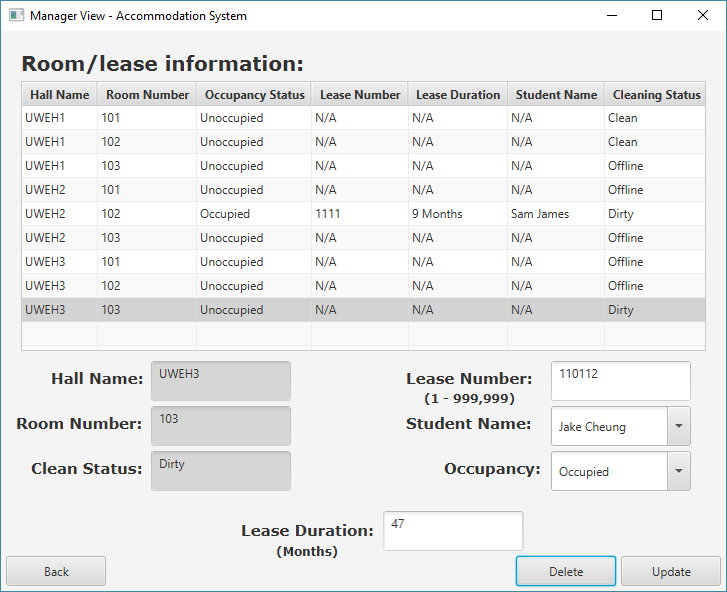


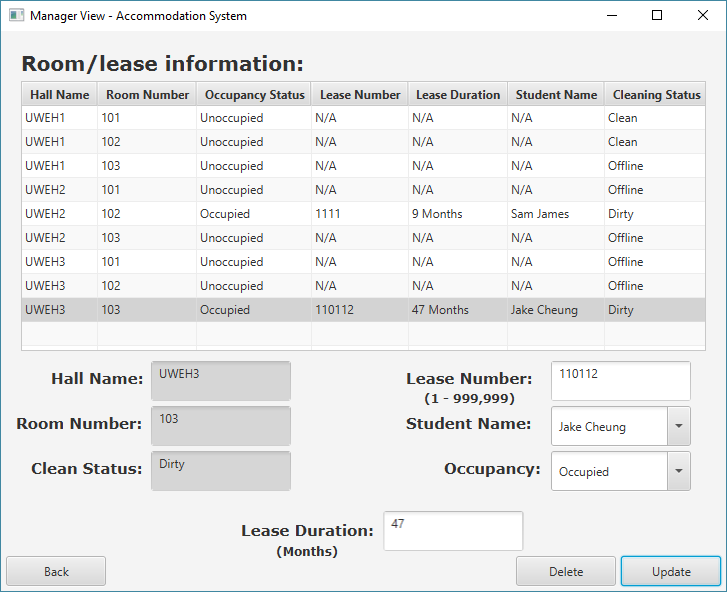
6, 7, 8 –



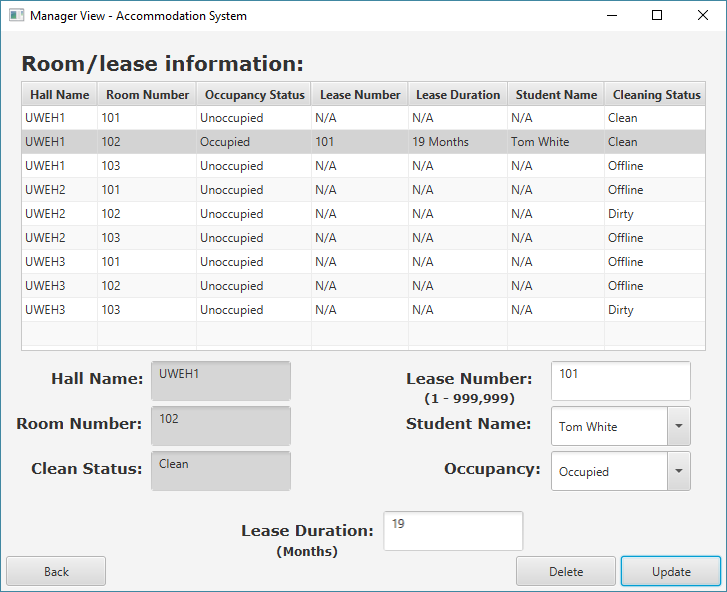


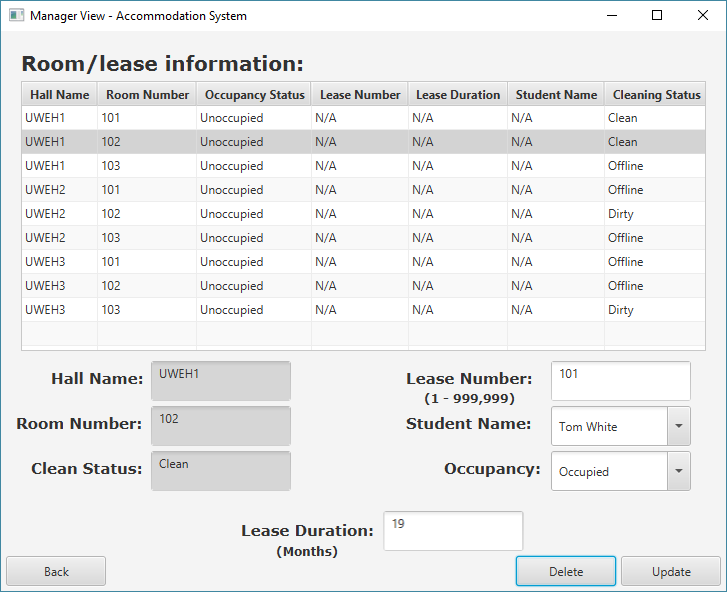
9-



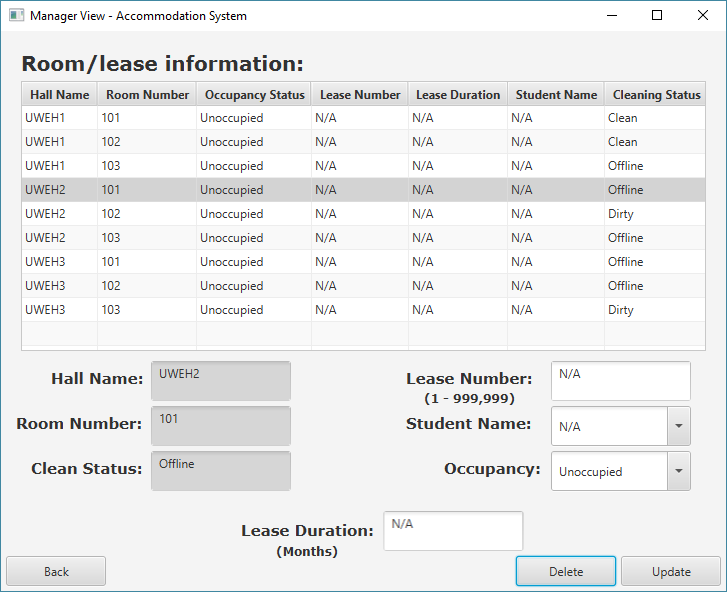


10 –



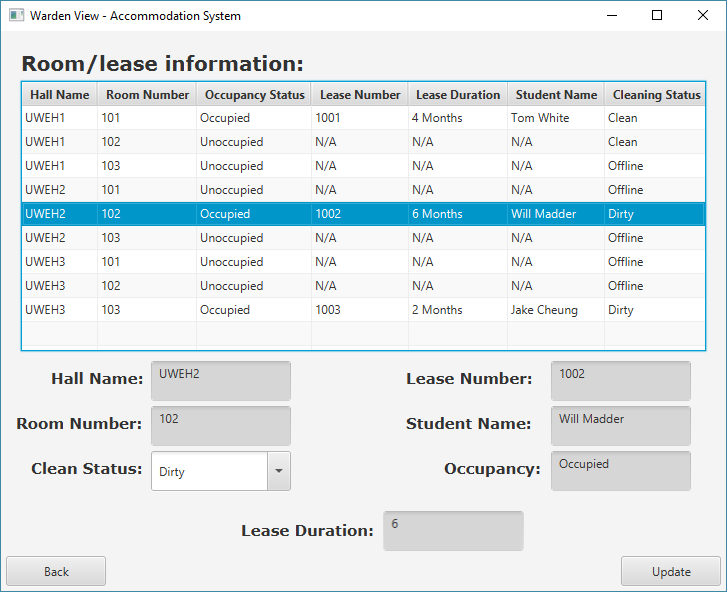


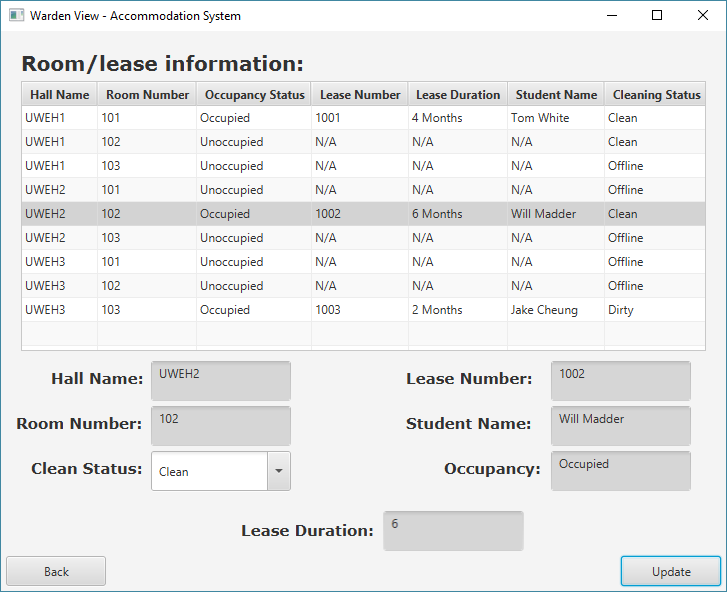
11 –



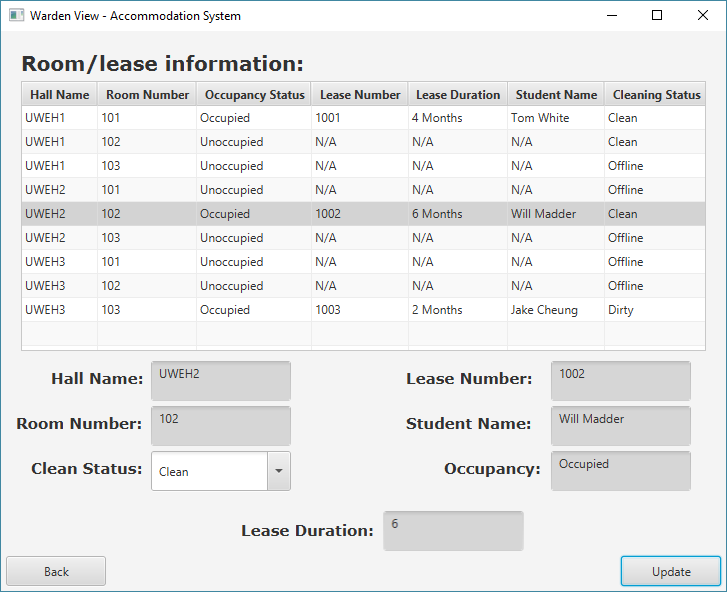


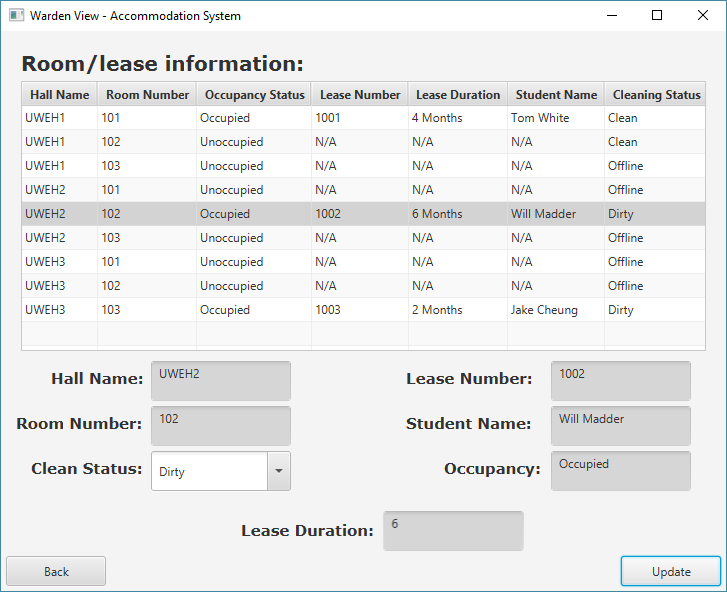
12 –



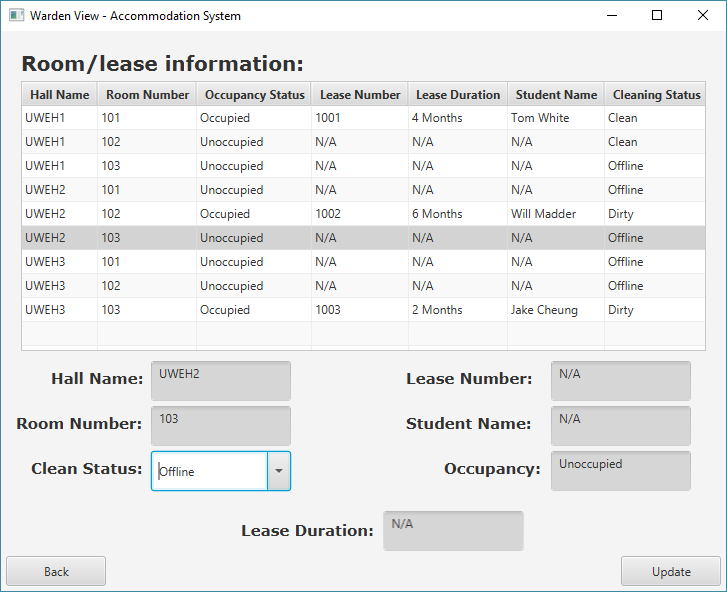


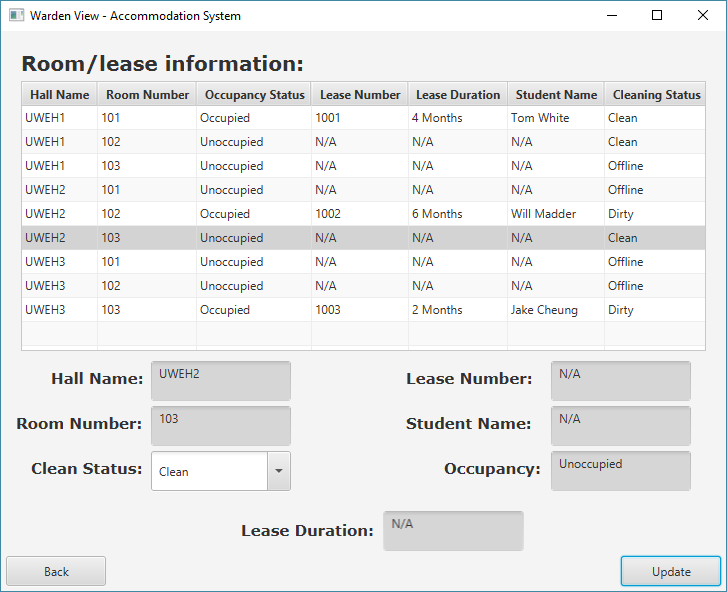
13 –



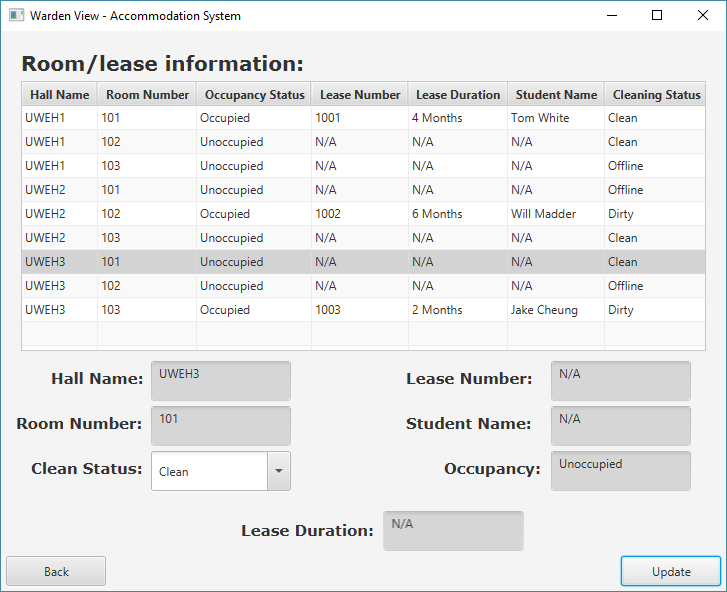


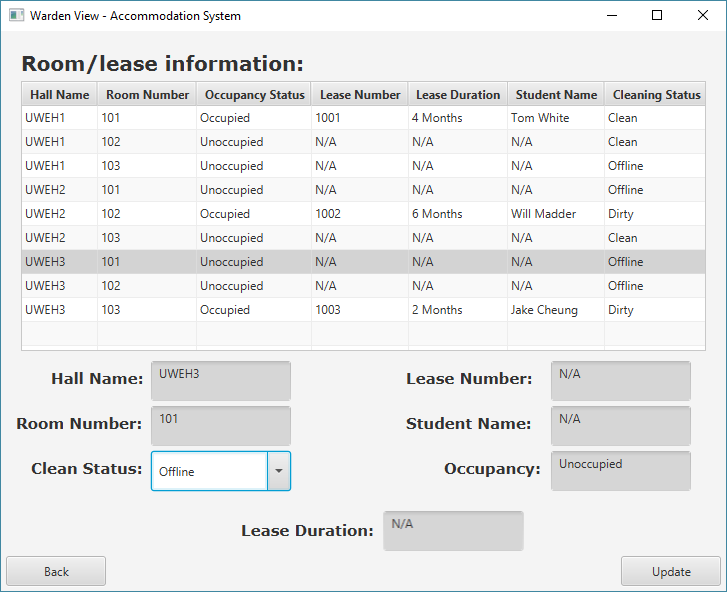
14 –



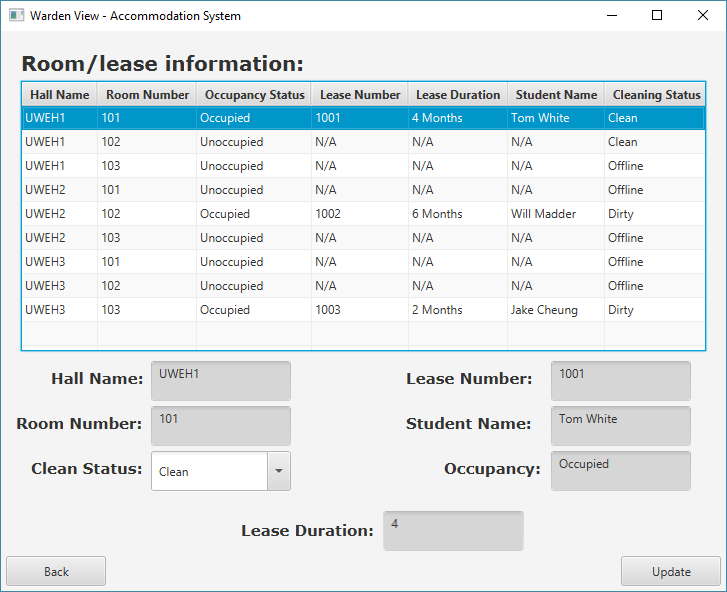


15 -



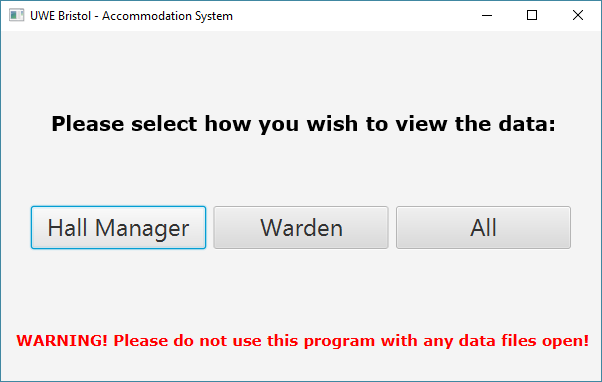


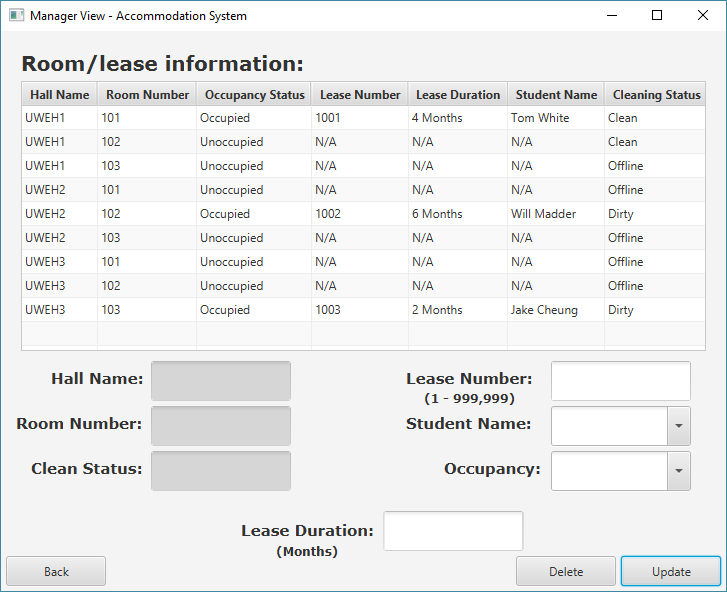
16 –



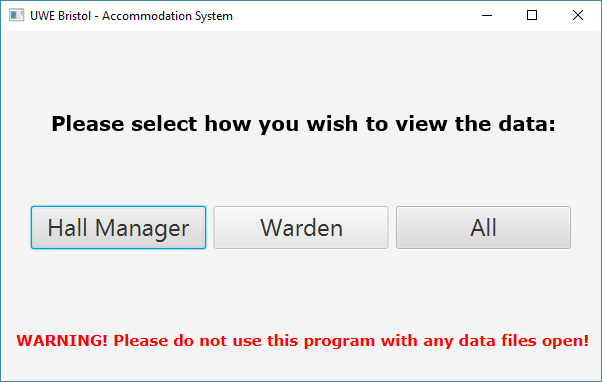


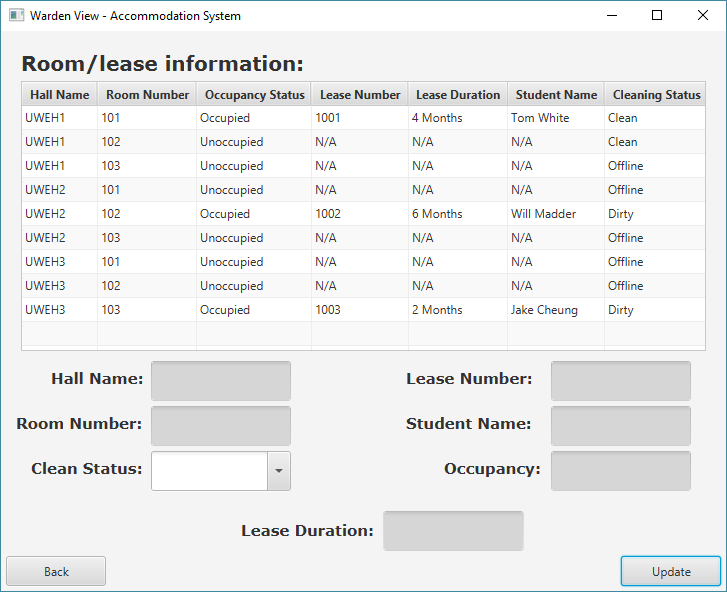
17 –



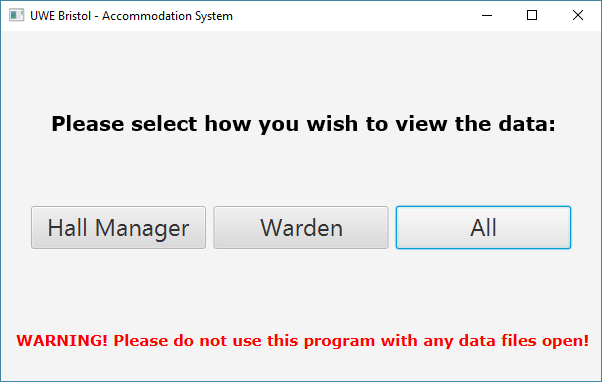


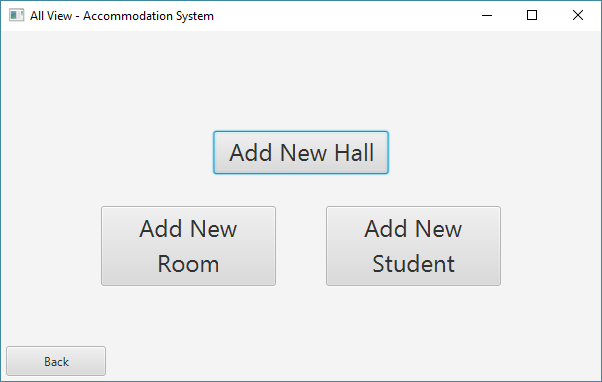
18 –



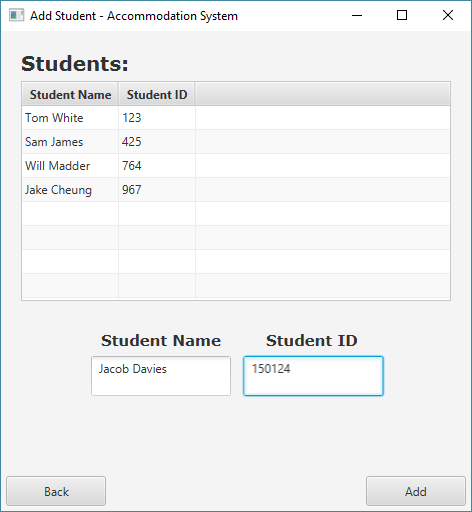


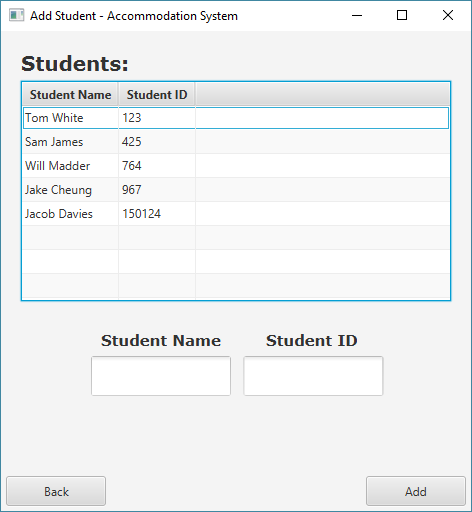
19 –



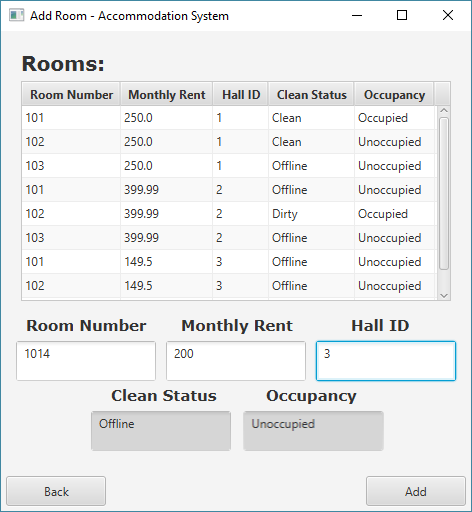


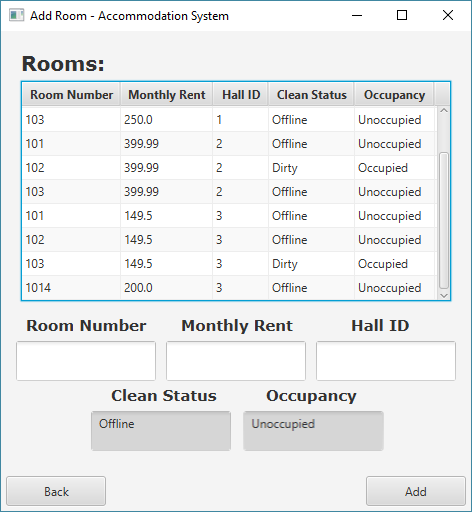
20 –



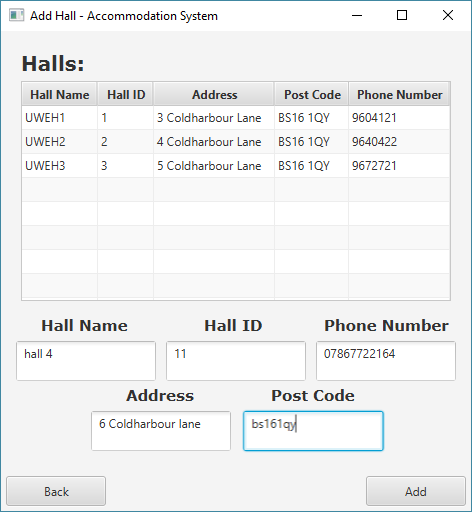


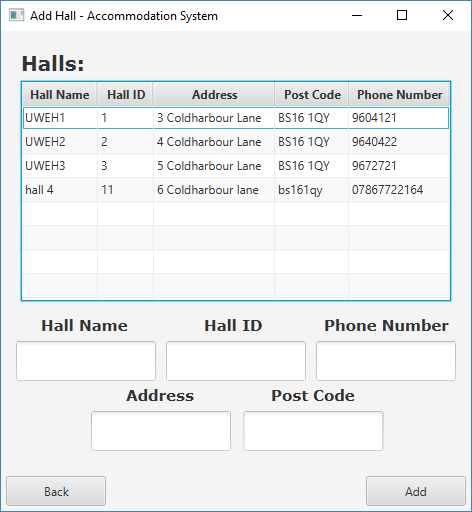
21 –



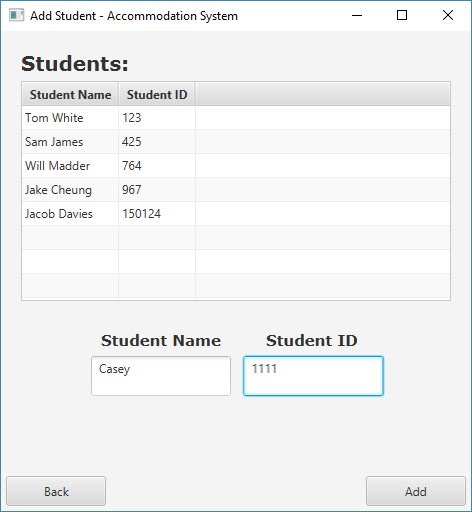


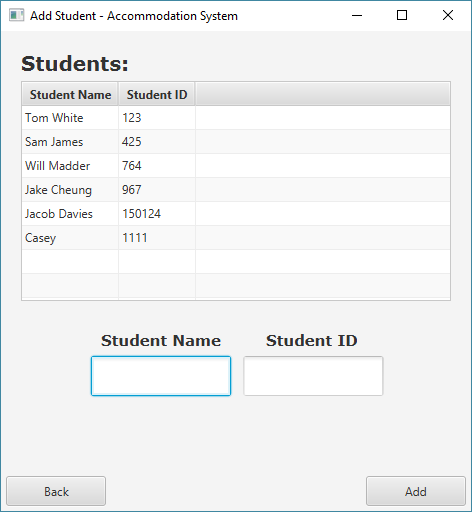
22 –



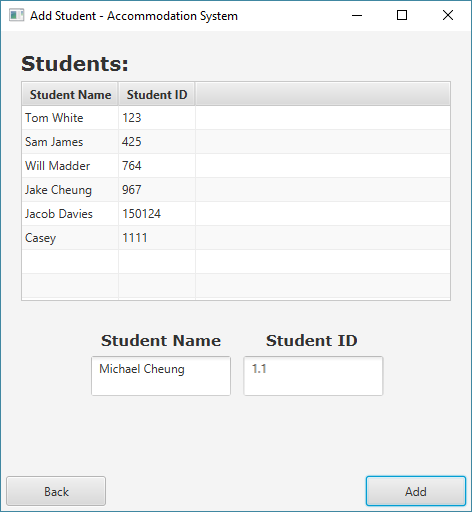


23 – note that the fix for this is included at the bottom of this document.



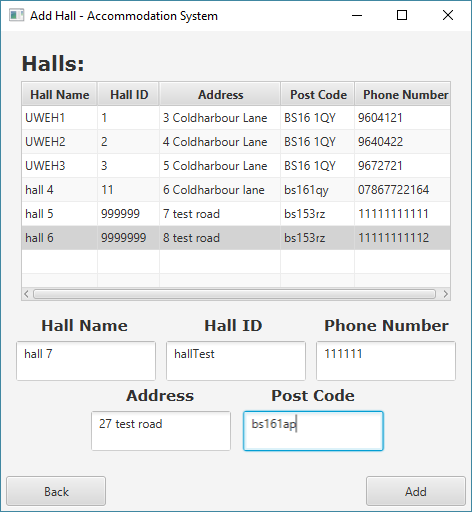


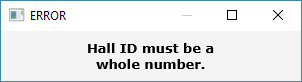
24 –



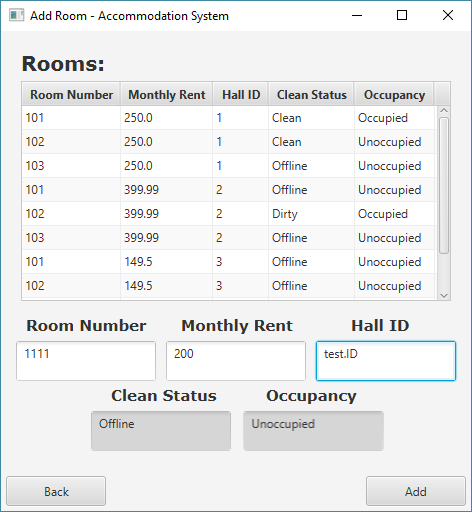


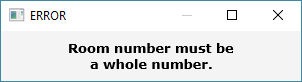
25 –



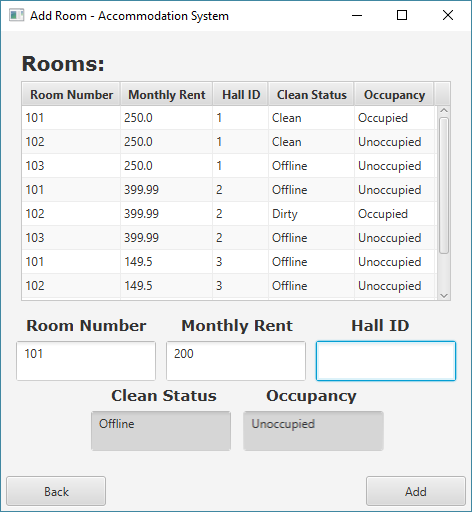


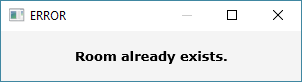
26 –



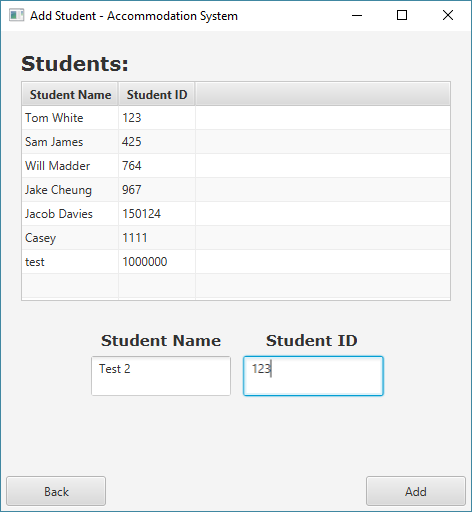


27 –



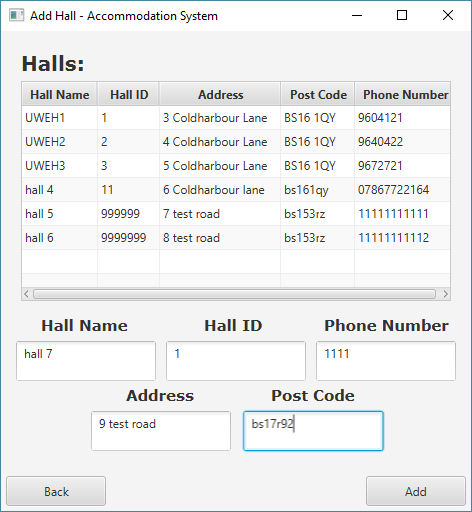


28 –



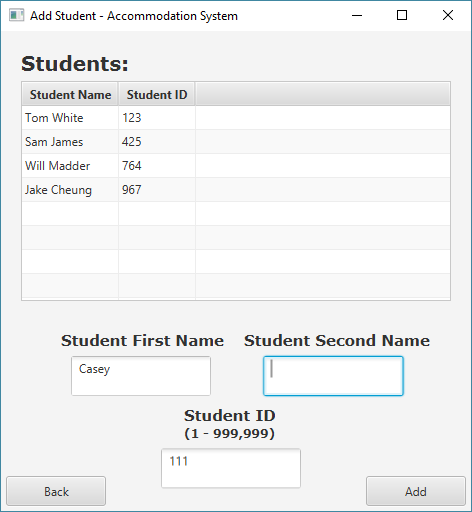


29 –





23 fixed –





**Junit testing –**

**Halls –**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package accommodation.application;  import java.lang.reflect.Field;  import org.junit.AfterClass;  import org.junit.BeforeClass;  import org.junit.Test;  import static org.junit.Assert.\*;  /\*\*  \*  \* @author Michael McCormick (15012271)  \*/  public class HallsTest {    public HallsTest() {  }    @BeforeClass  public static void setUpClass() {  }    @AfterClass  public static void tearDownClass() {  }  /\*\*  \* Test of getHallName method, of class Halls.  \*/  @Test  public void testGetHallName() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getHallName");  Halls instance = null;  final Field field = instance.getClass().getDeclaredField("hallName");  field.setAccessible(true);  field.set(instance, "UWEH1");  final String result = instance.getHallName();  assertEquals("Field wasn't retrieved properly", result, "UWEH1");  }  /\*\*  \* Test of getHallID method, of class Halls.  \*/  @Test  public void testGetHallID() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getHallID");  Halls instance = null;  final Field field = instance.getClass().getDeclaredField("hallID");  field.setAccessible(true);  field.set(instance, 1);  final int result = instance.getHallID();  assertEquals("Field wasn't retrieved properly", result, 1);  }  /\*\*  \* Test of getHallAddress method, of class Halls.  \*/  @Test  public void testGetHallAddress() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getHallAddress");  Halls instance = null;  final Field field = instance.getClass().getDeclaredField("hallAddress");  field.setAccessible(true);  field.set(instance, "3 Coldharbour Lane");  final String result = instance.getHallAddress();  assertEquals("Field wasn't retrieved properly", result, "3 Coldharbour Lane");  }  /\*\*  \* Test of getHallPostCode method, of class Halls.  \*/  @Test  public void testGetHallPostCode() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getHallPostCode");  Halls instance = null;  final Field field = instance.getClass().getDeclaredField("hallPostCode");  field.setAccessible(true);  field.set(instance, "BS16 1QY");  final String result = instance.getHallPostCode();  assertEquals("Field wasn't retrieved properly", result, "BS16 1QY");  }  /\*\*  \* Test of getPhoneNumber method, of class Halls.  \*/  @Test  public void testGetPhoneNumber() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getPhoneNumber");  Halls instance = null;  final Field field = instance.getClass().getDeclaredField("phoneNumber");  field.setAccessible(true);  field.set(instance, "9604161");  final String result = instance.getPhoneNumber();  assertEquals("Field wasn't retrieved properly", result, "9604161");  }    } |

**Lease –**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package accommodation.application;  import java.lang.reflect.Field;  import org.junit.AfterClass;  import org.junit.BeforeClass;  import org.junit.Test;  import static org.junit.Assert.\*;  /\*\*  \*  \* @author Michael McCormick (15012271)  \*/  public class LeaseTest {    public LeaseTest() {  }    @BeforeClass  public static void setUpClass() {  }    @AfterClass  public static void tearDownClass() {  }  /\*\*  \* Test of getLeaseNumber method, of class Lease.  \*/  @Test  public void testGetLeaseNumber() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getLeaseNumber");  Lease instance = null;  final Field field = instance.getClass().getDeclaredField("leaseNumber");  field.setAccessible(true);  field.set(instance, 1);  final int result = instance.getLeaseNumber();  assertEquals("Field wasn't retrieved properly", result, 1);  }  /\*\*  \* Test of getLeaseDuration method, of class Lease.  \*/  @Test  public void testGetLeaseDuration() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getLeaseDuration");  Lease instance = null;  final Field field = instance.getClass().getDeclaredField("leaseDuration");  field.setAccessible(true);  field.set(instance, 1);  final int result = instance.getLeaseDuration();  assertEquals("Field wasn't retrieved properly", result, 1);  }  /\*\*  \* Test of getHallID method, of class Lease.  \*/  @Test  public void testGetHallID() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getHallID");  Lease instance = null;  final Field field = instance.getClass().getDeclaredField("hallID");  field.setAccessible(true);  field.set(instance, 1);  final int result = instance.getHallID();  assertEquals("Field wasn't retrieved properly", result, 1);  }  /\*\*  \* Test of getRoomNumber method, of class Lease.  \*/  @Test  public void testGetRoomNumber() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getRoomNumber");  Lease instance = null;  final Field field = instance.getClass().getDeclaredField("roomNumber");  field.setAccessible(true);  field.set(instance, 1);  final int result = instance.getRoomNumber();  assertEquals("Field wasn't retrieved properly", result, 1);  }  /\*\*  \* Test of getStudentID method, of class Lease.  \*/  @Test  public void testGetStudentID() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getStudentID");  Lease instance = null;  final Field field = instance.getClass().getDeclaredField("studentID");  field.setAccessible(true);  field.set(instance, 1);  final int result = instance.getStudentID();  assertEquals("Field wasn't retrieved properly", result, 1);  }  /\*\*  \* Test of setLeaseNumber method, of class Lease.  \*/  @Test  public void testSetLeaseNumber() throws NoSuchFieldException, IllegalAccessException {  System.out.println("setLeaseNumber");  int leaseNumber = 1;  Lease instance = null;  instance.setLeaseNumber(leaseNumber);  final Field field = instance.getClass().getDeclaredField("leaseNumber");  field.setAccessible(true);  assertEquals("Fields didn't match", field.get(instance), 0);  }  /\*\*  \* Test of setLeaseDuration method, of class Lease.  \*/  @Test  public void testSetLeaseDuration() throws NoSuchFieldException, IllegalAccessException {  System.out.println("setLeaseDuration");  int leaseDuration = 1;  Lease instance = null;  instance.setLeaseDuration(leaseDuration);  final Field field = instance.getClass().getDeclaredField("leaseDuration");  field.setAccessible(true);  assertEquals("Fields didn't match", field.get(instance), 0);  }  /\*\*  \* Test of setStudentID method, of class Lease.  \*/  @Test  public void testSetStudentID() throws NoSuchFieldException, IllegalAccessException {  System.out.println("setStudentID");  int studentID = 1;  Lease instance = null;  instance.setStudentID(studentID);  final Field field = instance.getClass().getDeclaredField("studentID");  field.setAccessible(true);  assertEquals("Fields didn't match", field.get(instance), 0);  }    } |

**Room –**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package accommodation.application;  import java.lang.reflect.Field;  import org.junit.AfterClass;  import org.junit.BeforeClass;  import org.junit.Test;  import static org.junit.Assert.\*;  /\*\*  \*  \* @author Michael McCormick (15012271)  \*/  public class RoomTest {    public RoomTest() {  }    @BeforeClass  public static void setUpClass() {  }    @AfterClass  public static void tearDownClass() {  }  /\*\*  \* Test of getRoomNumber method, of class Room.  \*/  @Test  public void testGetRoomNumber() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getRoomNumber");  Room instance = null;  final Field field = instance.getClass().getDeclaredField("roomNumber");  field.setAccessible(true);  field.set(instance, 1);  final int result = instance.getRoomNumber();  assertEquals("Field wasn't retrieved properly", result, 1);  }  /\*\*  \* Test of getMonthlyRent method, of class Room.  \*/  @Test  public void testGetMonthlyRent() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getMonthlyRent");  Room instance = null;  final Field field = instance.getClass().getDeclaredField("monthlyRent");  field.setAccessible(true);  field.set(instance, 1.1);  final double result = instance.getMonthlyRent();  assertEquals("Field wasn't retrieved properly", result, 1.1);  }  /\*\*  \* Test of getHallID method, of class Room.  \*/  @Test  public void testGetHallID() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getHallID");  Room instance = null;  final Field field = instance.getClass().getDeclaredField("hallID");  field.setAccessible(true);  field.set(instance, 1);  final int result = instance.getHallID();  assertEquals("Field wasn't retrieved properly", result, 1);  }  /\*\*  \* Test of getCleanStatus method, of class Room.  \*/  @Test  public void testGetCleanStatus() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getCleanStatus");  Room instance = null;  final Field field = instance.getClass().getDeclaredField("cleanStatus");  field.setAccessible(true);  field.set(instance, "Clean");  final String result = instance.getCleanStatus();  assertEquals("Field wasn't retrieved properly", result, "Clean");  }  /\*\*  \* Test of getOccupancy method, of class Room.  \*/  @Test  public void testGetOccupancy() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getOccupancy");  Room instance = null;  final Field field = instance.getClass().getDeclaredField("occupancy");  field.setAccessible(true);  field.set(instance, "Occupied");  final String result = instance.getOccupancy();  assertEquals("Field wasn't retrieved properly", result, "Occupied");  }  /\*\*  \* Test of setCleanStatus method, of class Room.  \*/  @Test  public void testSetCleanStatus() throws NoSuchFieldException, IllegalAccessException {  System.out.println("setCleanStatus");  String cleanStatus = "Clean";  Room instance = null;  instance.setCleanStatus(cleanStatus);  final Field field = instance.getClass().getDeclaredField("cleanStatus");  field.setAccessible(true);  assertEquals("Fields didn't match", field.get(instance), "Clean");  }  /\*\*  \* Test of setOccupancy method, of class Room.  \*/  @Test  public void testSetOccupancy() throws NoSuchFieldException, IllegalAccessException {  System.out.println("setOccupancy");  String occupancy = "Occupied";  Room instance = null;  instance.setOccupancy(occupancy);  final Field field = instance.getClass().getDeclaredField("occupancy");  field.setAccessible(true);  assertEquals("Fields didn't match", field.get(instance), "Occupied");  }    } |

**Student –**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package accommodation.application;  import java.lang.reflect.Field;  import org.junit.AfterClass;  import org.junit.BeforeClass;  import org.junit.Test;  import static org.junit.Assert.\*;  /\*\*  \*  \* @author Michael McCormick (15012271)  \*/  public class StudentTest {    public StudentTest() {  }    @BeforeClass  public static void setUpClass() {  }    @AfterClass  public static void tearDownClass() {  }  /\*\*  \* Test of getStudentName method, of class Student.  \*/  @Test  public void testGetStudentName() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getStudentName");  Student instance = null;  final Field field = instance.getClass().getDeclaredField("studentName");  field.setAccessible(true);  field.set(instance, "John Smith");  final String result = instance.getStudentName();  assertEquals("Field wasn't retrieved properly", result, "John Smith");  }  /\*\*  \* Test of getStudentID method, of class Student.  \*/  @Test  public void testGetStudentID() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getStudentID");  Student instance = null;  final Field field = instance.getClass().getDeclaredField("studentID");  field.setAccessible(true);  field.set(instance, 1);  final int result = instance.getStudentID();  assertEquals("Field wasn't retrieved properly", result, 1);  }    } |

**Table –**

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package accommodation.application;  import java.lang.reflect.Field;  import org.junit.AfterClass;  import org.junit.BeforeClass;  import org.junit.Test;  import static org.junit.Assert.\*;  /\*\*  \*  \* @author Michael McCormick (15012271)  \*/  public class TableTest {    public TableTest() {  }    @BeforeClass  public static void setUpClass() {  }    @AfterClass  public static void tearDownClass() {  }  /\*\*  \* Test of getHallName method, of class Table.  \*/  @Test  public void testGetHallName() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getHallName");  Table instance = null;  final Field field = instance.getClass().getDeclaredField("hallName");  field.setAccessible(true);  field.set(instance, "UWEH1");  final String result = instance.getHallName();  assertEquals("Field wasn't retrieved properly", result, "UWEH1");  }  /\*\*  \* Test of getRoomNumber method, of class Table.  \*/  @Test  public void testGetRoomNumber() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getRoomNumber");  Table instance = null;  final Field field = instance.getClass().getDeclaredField("roomNumber");  field.setAccessible(true);  field.set(instance, 1);  final int result = instance.getRoomNumber();  assertEquals("Field wasn't retrieved properly", result, 1);  }  /\*\*  \* Test of getOccupancy method, of class Table.  \*/  @Test  public void testGetOccupancy() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getOccupancy");  Table instance = null;  final Field field = instance.getClass().getDeclaredField("occupancy");  field.setAccessible(true);  field.set(instance, "Occupied");  final String result = instance.getOccupancy();  assertEquals("Field wasn't retrieved properly", result, "Occupied");  }  /\*\*  \* Test of getCleanStatus method, of class Table.  \*/  @Test  public void testGetCleanStatus() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getCleanStatus");  Table instance = null;  final Field field = instance.getClass().getDeclaredField("cleanStatus");  field.setAccessible(true);  field.set(instance, "Clean");  final String result = instance.getCleanStatus();  assertEquals("Field wasn't retrieved properly", result, "Clean");  }  /\*\*  \* Test of getLeaseNumber method, of class Table.  \*/  @Test  public void testGetLeaseNumber() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getLeaseNumber");  Table instance = null;  final Field field = instance.getClass().getDeclaredField("leaseNumber");  field.setAccessible(true);  field.set(instance, "1001");  final String result = instance.getLeaseNumber();  assertEquals("Field wasn't retrieved properly", result, "1001");  }  /\*\*  \* Test of getLeaseDuration method, of class Table.  \*/  @Test  public void testGetLeaseDuration() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getLeaseDuration");  Table instance = null;  final Field field = instance.getClass().getDeclaredField("leaseDuration");  field.setAccessible(true);  field.set(instance, "9 Months");  final String result = instance.getLeaseDuration();  assertEquals("Field wasn't retrieved properly", result, "9 Months");  }  /\*\*  \* Test of getStudentName method, of class Table.  \*/  @Test  public void testGetStudentName() throws NoSuchFieldException, IllegalAccessException {  System.out.println("getStudentName");  Table instance = null;  final Field field = instance.getClass().getDeclaredField("studentName");  field.setAccessible(true);  field.set(instance, "John Smith");  final String result = instance.getHallName();  assertEquals("Field wasn't retrieved properly", result, "John Smith");  }  /\*\*  \* Test of setOccupancy method, of class Table.  \*/  @Test  public void testSetOccupancy() throws NoSuchFieldException, IllegalAccessException {  System.out.println("setOccupancy");  String occupancy = "Occupied";  Table instance = null;  instance.setOccupancy(occupancy);  final Field field = instance.getClass().getDeclaredField("occupancy");  field.setAccessible(true);  assertEquals("Fields didn't match", field.get(instance), "Occupied");  }  /\*\*  \* Test of setLeaseNumber method, of class Table.  \*/  @Test  public void testSetLeaseNumber() throws NoSuchFieldException, IllegalAccessException {  System.out.println("setLeaseNumber");  String leaseNumber = "1001";  Table instance = null;  instance.setLeaseNumber(leaseNumber);  final Field field = instance.getClass().getDeclaredField("leaseNumber");  field.setAccessible(true);  assertEquals("Fields didn't match", field.get(instance), "1001");  }  /\*\*  \* Test of setLeaseDuration method, of class Table.  \*/  @Test  public void testSetLeaseDuration() throws NoSuchFieldException, IllegalAccessException {  System.out.println("setLeaseDuration");  String leaseDuration = "9";  Table instance = null;  instance.setLeaseDuration(leaseDuration);  final Field field = instance.getClass().getDeclaredField("leaseDuration");  field.setAccessible(true);  assertEquals("Fields didn't match", field.get(instance), "9 Months");  }  /\*\*  \* Test of setStudentName method, of class Table.  \*/  @Test  public void testSetStudentName() throws NoSuchFieldException, IllegalAccessException {  System.out.println("setStudentName");  String studentName = "John Smith";  Table instance = null;  instance.setStudentName(studentName);  final Field field = instance.getClass().getDeclaredField("studentName");  field.setAccessible(true);  assertEquals("Fields didn't match", field.get(instance), "John Smith");  }  /\*\*  \* Test of setCleanStatus method, of class Table.  \*/  @Test  public void testSetCleanStatus() throws NoSuchFieldException, IllegalAccessException {  System.out.println("setCleanStatus");  String cleanStatus = "Clean";  Table instance = null;  instance.setCleanStatus(cleanStatus);  final Field field = instance.getClass().getDeclaredField("cleanStatus");  field.setAccessible(true);  assertEquals("Fields didn't match", field.get(instance), "Clean");  }    } |