# ITC205 – Professional Programming Practice

# Assessment Three UAT script

# John Galvin

# 11330960

# Scenario: Hotel Check in Use Case

The scenario scripts below are to test the Check in functionality of the program. Further details outlined below.

**Test Scripts:**

Testing the Hotel Check in Use cases for the following scenarios:

* Successful check in
* No booking found
* Already checked in
* Already checked out
* Room not ready

**Use Case:**

* Hotel Check in

**Test Components/Requirements:**

All of the scripts make use of the HotelHelper class to create a mock hotel, populate it with rooms and to also create a guest and a booking that will allow for the Check in scenarios to be tested.

**Actors:**

* Guest
* Hotel Employee

# Test Name: Successful Check in

**Use Case Tested:** Hotel Check In Use Case

**Test Description:** An existing guest who has a booking checks in to their room.

**Pre-conditions:**

* Guest has booked a room.
* Room.State is READY.
* Booking.State is PENDING

**Post-conditions:**

* Booking state updates to CHECKED\_IN
* Room state updated to OCCUPIED
* The active booking can be found by the hotel using the room number

**Notes:**

HotelHelper class is used but lines 40 – 42 are commented out because those are the features being tested.

The Hotel Helper creates a booking with the confirmation number 101301.

**Result:**

Warning – while the code functions correctly it appears that the CheckinCTL.completed() method is called twice, resulting in the “Checking in completed” message being displayed twice.

|  | TEST STEP | EXPECTED TEST RESULTS | Actual Results | P |
| --- | --- | --- | --- | --- |
|  | Start Program and initialize HotelHelper | Room : 101, SINGLE  Room : 201, DOUBLE  Room : 301, TWIN\_SHARE  Hotel Management System  Please select:  B: Book a Room  C: Check In  R: Record Service  D: Check Out    Q: Quit    Selection : |  | P |
|  | Input “C” to Check in – Press ENTER | Checking in  Enter confirmation Number: |  | P |
|  | Input “101301” – Press ENTER | Twin share 301 is booked from 01-01-0001 for 1 nights by Fred.  Visa Credit card number: 2  Confirmation Number : 101301  Confirm hotel.checkin? (Y/N) : |  | P |
|  | Input “y” – Press ENTER | Checking in completed  Hit <enter> to continue |  | P |
|  | Press Enter | Hotel Management System  Please select:  B: Book a Room  C: Check In  R: Record Service  D: Check Out  Q: Quit  Selection : |  | F |

# Test Name: Booking not found

**Use Case Tested:** Hotel Check in

**Test Description:** Guest attempts to check in to a room but their confirmation number does not exist.

**Pre-conditions:**

* Booking not made
* Confirmation number does not exist

**Post-conditions:**

**Notes:** HotelHelper used to initialize a hotel instance.

**Result:**

Warning - Program compiles and runs successfully but program is stuck in a loop unless they can enter an existing confirmation number. No other exit strategies from this point.

|  | TEST STEP | EXPECTED TEST RESULTS | Actual Results | P |
| --- | --- | --- | --- | --- |
|  | Start Program and initialize HotelHelper | Room : 101, SINGLE  Room : 201, DOUBLE  Room : 301, TWIN\_SHARE  Hotel Management System  Please select:  B: Book a Room  C: Check In  R: Record Service  D: Check Out    Q: Quit    Selection : |  | P |
|  | Input “C” – press ENTER | Checking in  Enter confirmation Number: |  | P |
|  | Input “123456” – press ENTER | No hotel.booking for confirmation number 123456 found  Enter confirmation number: |  | P |

# Test Name: Already Checked In

**Use Case Tested:** Hotel Check In

**Test Description:** Guest attempts to check in using a booking confirmation number that is already checked in.

**Pre-conditions:**

* Booking state is CHECKED\_IN
* Room state is OCCUPIED
* The active booking can be found by the hotel using the room number

**Post-conditions:**

* xc

**Notes:**

HotelHelper used to initialize hotel, create guest, rooms and booking – the code within the hotel helper that checks in the room is NOT commented out to set up the scenario required for testing.

**Result:**

Warning - Program compiles and runs successfully but program is stuck in a loop unless they can enter an existing confirmation number. No other exit strategies from this point.

|  | TEST STEP | EXPECTED TEST RESULTS | Actual Results | P |
| --- | --- | --- | --- | --- |
|  | Start Program and initialize HotelHelper | Room : 101, SINGLE  Room : 201, DOUBLE  Room : 301, TWIN\_SHARE  Hotel Management System  Please select:  B: Book a Room  C: Check In  R: Record Service  D: Check Out    Q: Quit    Selection : |  | P |
|  | Input “C” – press ENTER | Checking in  Enter confirmation Number: |  | P |
|  | Input “101301” – press ENTER | Booking 101301 has already been checked in  Enter confirmation number: |  | P |

# Test Name: Already Checked Out

**Use Case Tested:** Hotel Check in

**Test Description:** Guest attempts to check out of hotel using a booking that is already checked out.

**Pre-conditions:**

* Booking.State.CHECKED\_OUT

**Post-conditions:**

* condition

**Notes:**

HotelHelper used to initialize hotel, create guest, rooms and booking – added additional code to checkout the booking as well to set up scenario for testing. Hotel helper checks the guest into room with ID: 301

hotel.checkout(301); added to HotelHelper class

**Result:**

Initial Test Failure (results shown below) – the Room class was missing the functionality to add a booking into the list of bookings associated with a room. This resulted in an error calling the hotel.Checkout(roomID) method as there were no association between the Room and Booking classes.

Added in bookings.add(booking); under Room.book() to allow for successful testing.

Successful test – Warning: Program will remain stuck in loop unless a confirmation number can be entered.

|  | TEST STEP | EXPECTED TEST RESULTS | Actual Results | P |
| --- | --- | --- | --- | --- |
|  | Start Program and initialize HotelHelper | Room : 101, SINGLE  Room : 201, DOUBLE  Room : 301, TWIN\_SHARE  Hotel Management System  Please select:  B: Book a Room  C: Check In  R: Record Service  D: Check Out    Q: Quit    Selection : |  | P |
|  | Input “C” – press ENTER | Checking in  Enter confirmation number: |  | P |
|  | Input “101301” – Press enter | Booking 101301 has already been checked out  Enter confirmation number: |  | P |

# Failure of Test - Already Checked Out

|  | TEST STEP | EXPECTED TEST RESULTS | Actual Results | P |
| --- | --- | --- | --- | --- |
|  | Start Program and initialize HotelHelper | Room : 101, SINGLE  Room : 201, DOUBLE  Room : 301, TWIN\_SHARE  Hotel Management System  Please select:  B: Book a Room  C: Check In  R: Record Service  D: Check Out    Q: Quit    Selection : |  | F |

# Test Name: Room not ready

**Use Case Tested:** Hotel check in

**Test Description:** Guest attempts to check in to a room that is already occupied.

**Pre-conditions:**

* Room.State is OCCUPIED
* Booking.State is PENDING

**Post-conditions:**

* condition

**Notes:**

HotelHelper used to initialize hotel, create guest, rooms and booking. Additional code added to set room.state to OCCUPIED.

room.state = Room.State.OCCUPIED;

commented out the code in HotelHelper checking in the new booking.

**Result:**

Warning – correct message is displayed but program remains trapped in a loop unless a real confirmation number is entered.

|  | TEST STEP | EXPECTED TEST RESULTS | Actual Results | P |
| --- | --- | --- | --- | --- |
|  | Start Program and initialize HotelHelper | Room : 101, SINGLE  Room : 201, DOUBLE  Room : 301, TWIN\_SHARE  Hotel Management System  Please select:  B: Book a Room  C: Check In  R: Record Service  D: Check Out    Q: Quit    Selection : |  | P |
|  | Input “C” – press ENTER | Checking in  Enter confirmation number: |  | P |
|  | Input “101301” – Press enter | Room is not ready, sorry  Enter confirmation number: |  | P |