**Hotel Management System Overhaul**



***Use Case Specification Document***

**Check In/Out**

**Version No. 2**

**Project Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version**  **#** | **Date** | **Revision Author** | **Description of Revision** |
| 1 | 6/26/20 | Monier Abdullah | Initial Version |
| 2 | 7/29/20 | Monier Abdullah | Updated minor details that allow employees to check guests into reservations instead of guests checking into their own reservations. |

|  |
| --- |
| **Table of Contents** |

[**Introduction**](#_heading=h.gjdgxs) **3**

[**Use Case Information**](#_heading=h.2et92p0) **4**

[Actors](#_heading=h.tyjcwt) 4

[Use Case Interaction](#_heading=h.1t3h5sf) 4

[**Trigger**](#_heading=h.2s8eyo1) **5**

[**Pre-condition(s)**](#_heading=h.17dp8vu) **5**

[The Employee must be successfully able to access the hotel management system (via Access Hotel System use case) and verify a reservation exists.](#_heading=h.26in1rg) 5

[**Post-condition(s)**](#_heading=h.lnxbz9) **5**

[The status of a reservation is changed](#_heading=h.1ksv4uv) 5

[**Use Case Swimlane (Activity) Diagram**](#_heading=h.44sinio) **5**

[**Main/Basic Flow(s) of Events (Happy Path)**](#_heading=h.2jxsxqh) **6**

[Access a reservation](#_heading=h.z337ya) 6

[**Alternate/Exception Flow of Events**](#_heading=h.3j2qqm3) **6**

[**Assumptions/Business Rules including Non-Functional Requirements**](#_heading=h.2xcytpi) **6**

[**Use Case Specification Review and Signoff**](#_heading=h.2bn6wsx) **6**

# **Introduction**

This document captures detailed functional and non-functional BUSINESS requirements. Technical or application IT requirements should not be detailed here. A separate Use Case Summary document ties ALL the individual use cases together. First create the Use Case Summary document using application decomposition. Then increase the detail by creating the individual use case specifications – be careful not to create too many or not create enough use cases.

The Check In/Check Out use case enables hotel guests to check into an existing hotel reservation. At check in, employees will be able to process guest payments. Also, employees will be able to check out guests of their reservation.

# **Use Case Information**

## **Actors**

An actor is someone or something (e.g. application system) outside the system or business that interacts with the application. For every Use Case, there must be at least one Main Actor and zero or more Secondary Actors. Actors should be a person, system, or time.

|  |  |  |
| --- | --- | --- |
| Actor Name | Role | Description |
| Employee | Main | Primary Actor: Interacts with the system to check guest into their reservation, view reservation information, process payments, and check out of the reservation |
| System | Secondary | Secondary Actor: Interacts with the system to store the reservation check in/out status and payment status. |
| Hotel Management System | Secondary | Secondary Actor: Interacts with the system to store reservation check in/out and balance information. |

## **Use Case Interaction**

**Successor Use Cases**

* N/A

**Predecessor Use Cases**

* Access Hotel System
  + Employees must login to their account to begin check in/out procedure.
* Reservation
  + A reservation needs to exist before checking in/out.

# **Trigger**

* Employee logs into the system in order to check into or check out of a guest reservation

# **Pre-condition(s)**

## **The Employee must be able to access the hotel management system (via Access Hotel System use case).**

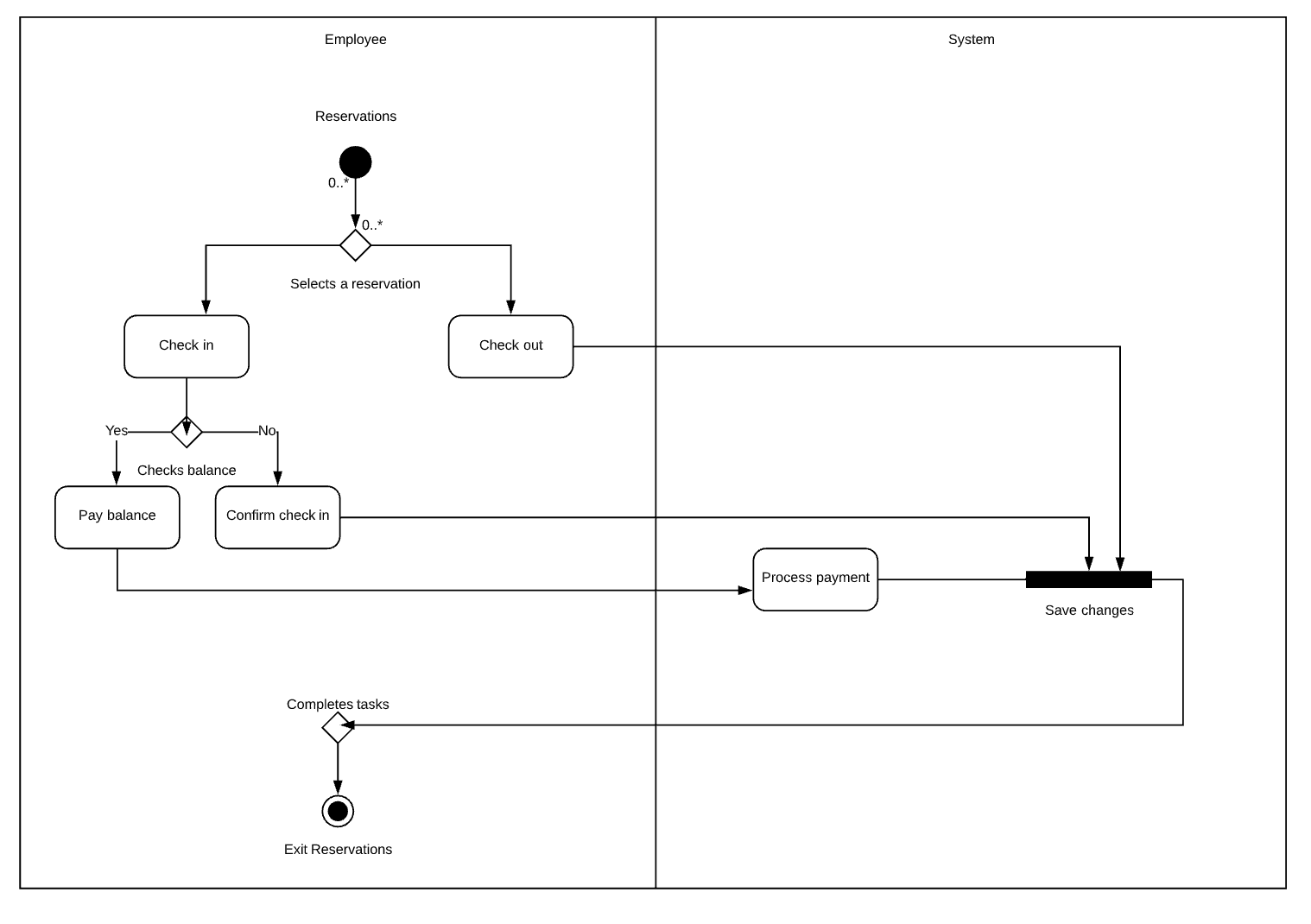
## **The Employee must locate a reservation.**

# **Post-condition(s)**

## **The status of a reservation is changed**

* + 1. **Reservation status is changed to checked in.**
    2. **Reservation payment status is changed to paid.**
    3. **Reservation status is changed to checked out.**

# **Use Case Swimlane (Activity) Diagram**



# **Main/Basic Flow(s) of Events (Happy Path)**

## **Access a reservation**

* + 1. The customer selects to manage a reservation
    2. Selects a reservation and changes it to:
       1. Check in
          1. Process payment if there is a balance.
       2. Check out
          1. Check out of reservation.
    3. The status of the reservation is saved.

# **Alternate/Exception Flow of Events**

**N/A**

# **Assumptions/Business Rules including Non-Functional Requirements**

**N/A**

# **Use Case Specification Review and Signoff**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review and Signoff of the Use Case Specification | | | | |
| Name | Project Team Role | Signature | Date | |
| Peter Schubert | Team Lead/ Software Developer | PS | 6/26/20 | |
| Jeffrey Fishman | Software Developer | JF | 6/26/20 | |
| Monier Abdullah | Software Developer | MA | 6/26/2020 | |
| Khalid Saeed | Software Developer | KS | 6/27/20 | |