**Hotel Management System Overhaul**

Use Case Specification Document

**Reservations**

**Version No. 2**

**Project Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Date** | **Revision Author** | **Description of Revision** |
| 1 | 6/25/20 | Jeffrey Fishman | Initial Version |
| 2 | 8/10/20 | Jeffrey Fishman | Include View Reservations flow |

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

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

[**Use Case Information**](#_heading=h.k4m7afqelf8r) **4**

[Actors](#_heading=h.1fob9te) 4

[Use Case Interaction](#_heading=h.2et92p0) 5

[**Trigger**](#_heading=h.s2qthit4vaeo) **5**

[**Pre-condition(s)**](#_heading=h.1lnlm89pg8yd) **5**

[**Post-condition(s)**](#_heading=h.b9x0dco6qn5) **5**

[**Use Case Swimlane (Activity) Diagram**](#_heading=h.v9rleatvb3l3) **6**

[**Main/Basic Flow(s) of Events (Happy Path)**](#_heading=h.kt5fuo5cx8ki) **6**

[**Alternate/Exception Flow of Events**](#_heading=h.7pwv2is0vuw7) **6**

[**Assumptions/Business Rules including Non-Functional Requirements**](#_heading=h.kus4so4fr3pf) **8**

[**Use Case Specification Review and Signoff**](#_heading=h.24vuzpkqlumd) **8**

# 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.

Write a single paragraph describing the purpose of the specific use case in the Introduction.

The Reservation use case handles the cases of a Customer looking at and purchasing hotel room or event reservations, Employees coming in to edit or remove existing reservations, or third-party systems requesting to add a reservation someone made on a different platform. It’s goal is to allow Customers to both view, but take no action, and also view then book available rooms, as well as to track room and hotel popularity and purchase rewards to a Customer account. Employees deleting an existing reservation should undo the whole transaction so the Customer gets their money and rewards back, and third party reservations will ask if a Customer would like to create an account for rewards tracking. All purchases will ask if the Customer would like to upgrade.

# 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 |
| Customer | Primary | Primary Actor: interacts with the system to login, book stays, view and manage individual reward points, edit name and location. Also interacts with Employee to modify the reservation |
| Employee | Primary | Primary Actor: interacts with system to cancel and review reservations, create accounts for customers, view summary reports and hotel informatics, and manage the hotel and other various functions |
| Third Party Booking Website | Primary | Primary Actor: Interacts with the system to make requests for booking from third party websites to existing or new customers and create new users. |
| System | Secondary | Secondary Actor: interacts with the system to store the necessary data about the customer, employees, hotel, and reservation |
| Hotel Management System | Secondary | Secondary Actor: Interacts with the system to store, collect, or manipulate data within the system’s data stores present back to the Customers and Employees. |

## Use Case Interaction

How does this use case relate to other use cases? List predecessor and successor use cases.

**Successor Use Cases**

* Check-in / Check-out
  + A check-in or check-out to a hotel room requires a successful reservation and payment to occur first

**Predecessor use Cases**

* Access Hotel System
  + User needs to successfully log in in order to view, reserve, and perform transactions for creating or modifying reservations

# Trigger

* User enters into the system and views / selects a hotel to reserve, or selects an existing reservation to modify
* A third-party reservation is queued in the system

# Pre-condition(s)

What use cases or other pre-conditions must be met before use can initiate?

## User must be successfully logged in to the system

* + 1. Without a successful login, a user can’t view hotels to reserve or existing reservations

## System must be fully configured with information about each hotel

* + 1. There must be appropriate hotels, hotel rooms, room rewards, and hotel events for each location, and each location must be valid

# Post-condition(s)

What are ALL the possible output states upon completion of the use case flows?

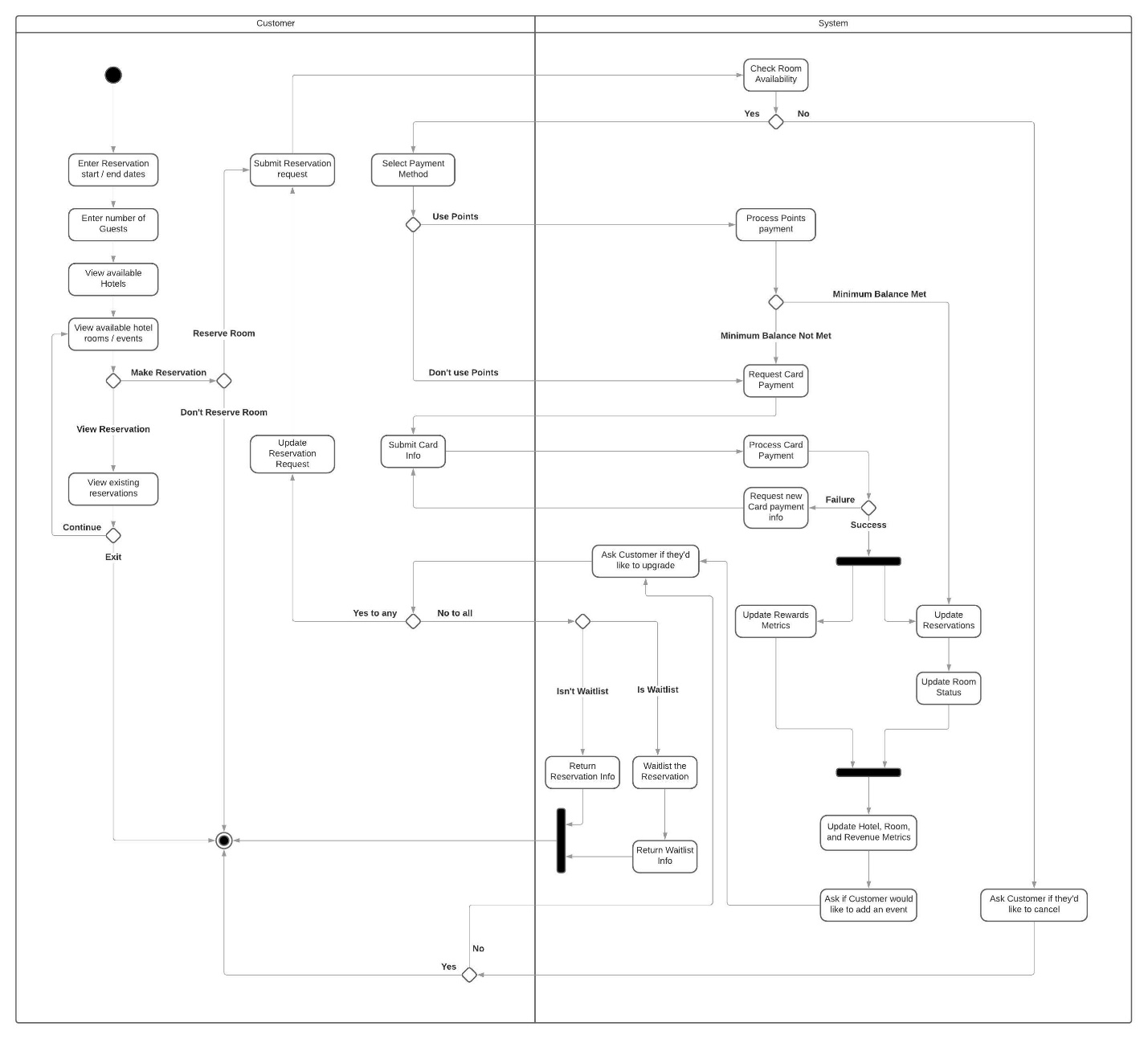
## A new reservation and reservation number are created, partially paid

* + 1. Room status changes from available to reserved, and additional payment is required to allow check-in (Check in / Check out use case)
  1. **A new reservation and reservation number are created, fully paid**
     1. Room status changes from available to reserved, and check-in is allowed (Check in / Check out use case)
  2. **An existing reservation is modified or deleted by employee**
     1. An existing reservation’s check-in / check-out dates are modified
     2. An existing reservation’s number of guests is modified
  3. **An existing reservation’s info is returned**
  4. **A third-party reservation is imported to the system**
     1. Third party reservation information is received and imported to the system
     2. Third-party customer creates an account for their imported reservation (Invokes Access Hotel System use case)
  5. .A new or imported reservation is waitlisted on the system

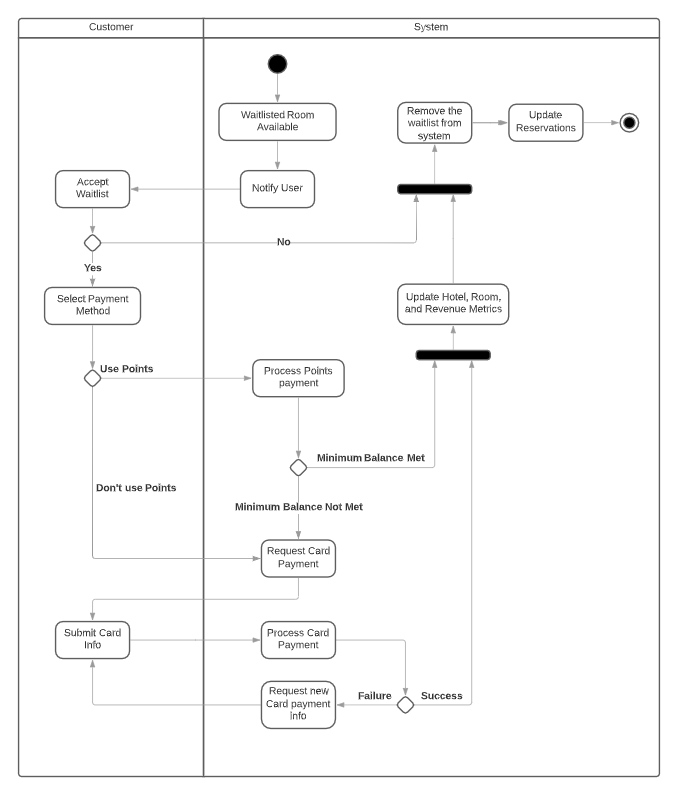
# Use Case Swimlane (Activity) Diagram

Draw diagram(s) that cover ALL main and alternate flows.

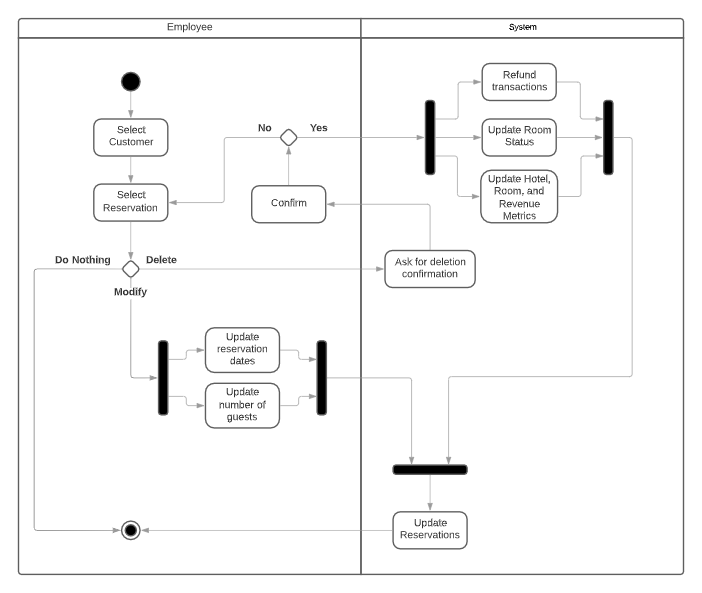
**Primary Customer flows:**



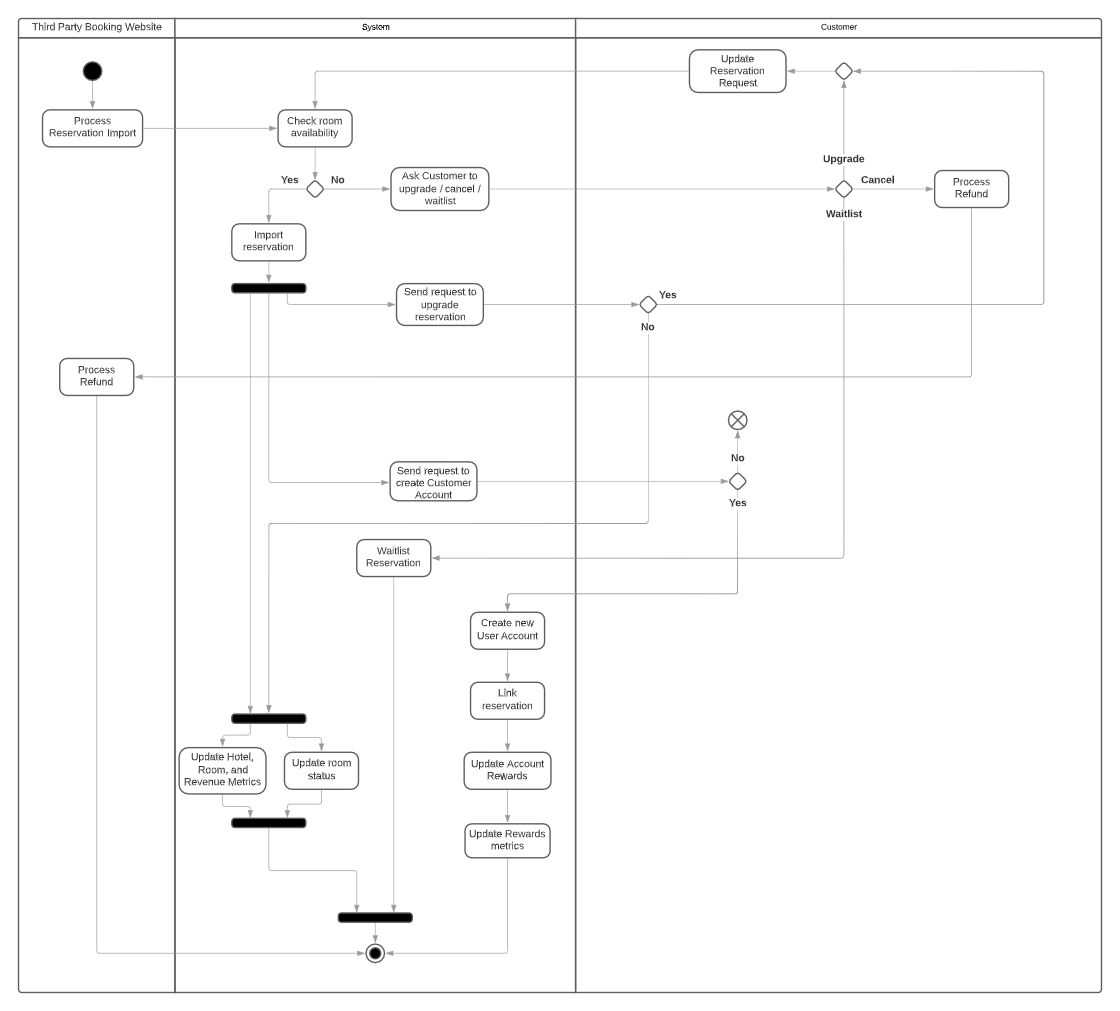
**Waitlist notification flows:**



**Administrator flows:**



**Third-Party Booking Website flows:**



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

For each main flow (usually ONE flow) write the list of steps that occur – describe WHAT occurs not HOW to do it!

* 1. **Create a reservation for a room or event**
     1. Customer provides reservation start and end dates
     2. Customer provides number of guests for the room or event
     3. Customer selects an available hotel and room / event to reserve
     4. Customer submits reservation for payment processing with Points, Card, or a combination of both
     5. System saves the reservation and a reservation number is returned to the customer
        1. Room status is updated from Available to Reserved
     6. If paid with Card, system awards Points to Customer
     7. Hotel, room, rewards, and revenue metrics are updated
     8. Customer is asked if they’d like to upgrade the room and/or add an event for an additional cost
  2. **View hotel information**
     1. User provides desired check-in / check-out dates
     2. User provides desired number of guests
     3. View available hotels
     4. View room / event information
        1. Availability
        2. Guest capacity
        3. Pricing
        4. Rewards if reserved
  3. **View reservation info**
     1. User selects to view their own existing reservations
     2. System returns reservation info
        1. Check in / out dates
        2. Number of Guests
        3. Room info
        4. Pricing
        5. Rewards
  4. **Modify a reservation for a room or event**
     1. Employee and/or customer selects an existing reservation
     2. Employee modifies number of guests up to the original reservation’s guest limit
     3. Employee and/or customer modifies check in / check out dates
     4. System saves the changes to the reservation
  5. **Delete a reservation for a room or event**
     1. Employee and/or customer selects an existing reservation to remove
     2. Employee and/or customer confirms they would like to delete the reservation
     3. System refunds reservation’s transactions /points accumulated
     4. System saves the changes to the reservation
        1. System adjusts the total number of available rooms for the room type cancelled
     5. Hotel, room, rewards, and revenue metrics are updated
  6. **Import a third-party booking to the system**
     1. Third-Party Booking Website submits reservation information to the System
     2. System checks for room / event availability, makes modifications, cancellation, and/or creation of account based on the file request
     3. System saves the changes to reservations
        1. System adjusts the total number of available room type
     4. Hotel, room, and revenue metrics are updated
     5. If customer made an account, rewards metrics are updated

# Alternate/Exception Flow of Events

For each alternative flow (can be zero or more) write the list of steps that occur – describe WHAT occurs not HOW to do it!

* 1. **Waitlist a new or imported reservation**
     1. System asks Customer if they would like to upgrade or cancel their reservation
     2. If not cancelled and not imported, System queues a Customer’s reservation to a waitlist on a room
     3. Notify customers of waitlisting
     4. When the room’s status changes to available, notify the waitlisted Customers
     5. Customer accepts or deletes the reservation from the notice
     6. If reserving, system requests payment with Points, Card, or a combination of both
     7. System updates reservation and waitlist information
     8. Hotel, room, rewards, and revenue metrics are updated
  2. **Payment for a reservation fails**
     1. System notifies Customer that payment failed is invalid
     2. Customer is asked to enter payment information again

# Assumptions/Business Rules including Non-Functional Requirements

Be sure to number the assumption/business rules to allow easy reference to them. Business rules will be where non-functional requirements are recorded – have a way to specifically identify 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 |
| Jeffrey Fishman | Team Lead / Software Developer | JF | 8/10/20 |
| Monier Abdullah | Software Developer | MA | 6/27/2020 |
| Khalid Saeed | Software Developer | KS | 6/27/20 |