**Rewards**



***Use Case Specification Document***

**Rewards**

**Version No. 1**

**Project Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version**  **#** | **Date** | **Revision Author** | **Description of Revision** |
| 1 | 6/26/20 | Monier Abdullah | Initial Version |
|  |  |  |  |

|  |
| --- |
| **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**

[**Post-condition(s)**](#_heading=h.1ksv4uv) **5**

[**Use Case Swimlane (Activity) Diagram**](#_heading=h.1y810tw) **5**

[**Main/Basic Flow(s) of Events (Happy Path)**](#_heading=h.4i7ojhp) **6**

[Customer books a reservation](#_heading=h.2xcytpi) 6

[**Alternate/Exception Flow of Events**](#_heading=h.3whwml4) **6**

[**Assumptions/Business Rules including Non-Functional Requirements**](#_heading=h.3as4poj) **6**

[**Use Case Specification Review and Signoff**](#_heading=h.2p2csry) **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.

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

Rewards are applied to a user’s account each time they book a reservation. The user can earn rewards and apply them to their reservation for discounts and add events or packages to 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 |
| Customer | Main | Primary Actor: interacts with system to earn rewards points and apply them to their reservation. |
| System | Secondary | Secondary Actor: interacts with the system to check the invoice details for rewards calculations, check amount of reward points, and checks available discounts and event packages. |
| Hotel Management System | Secondary | Interacts with the system to store new reward points and apply reward points to invoice |

## **Use Case Interaction**

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

**Successor Use Cases**

* Rewards
  + Based on the rewards, the user is able to apply discounts and/or event packages before processing a payment.

**Predecessor use Cases**

* Access Hotel System
  + Customer needs to login in order to view and apply reward points to their reservation.

# **Trigger**

* Customer logs into their account and selects View and Manage Rewards.

# **Pre-condition(s)**

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

## **The Customer must be successfully able to access their account via Access Hotel System Use Case.**

## **Customer must book a reservation to earn reward points.**

## **Customer must book a reservation to use reward points towards discounts/event packages.**

# **Post-condition(s)**

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

## **Reward points are added to the customer’s account after booking a reservation**

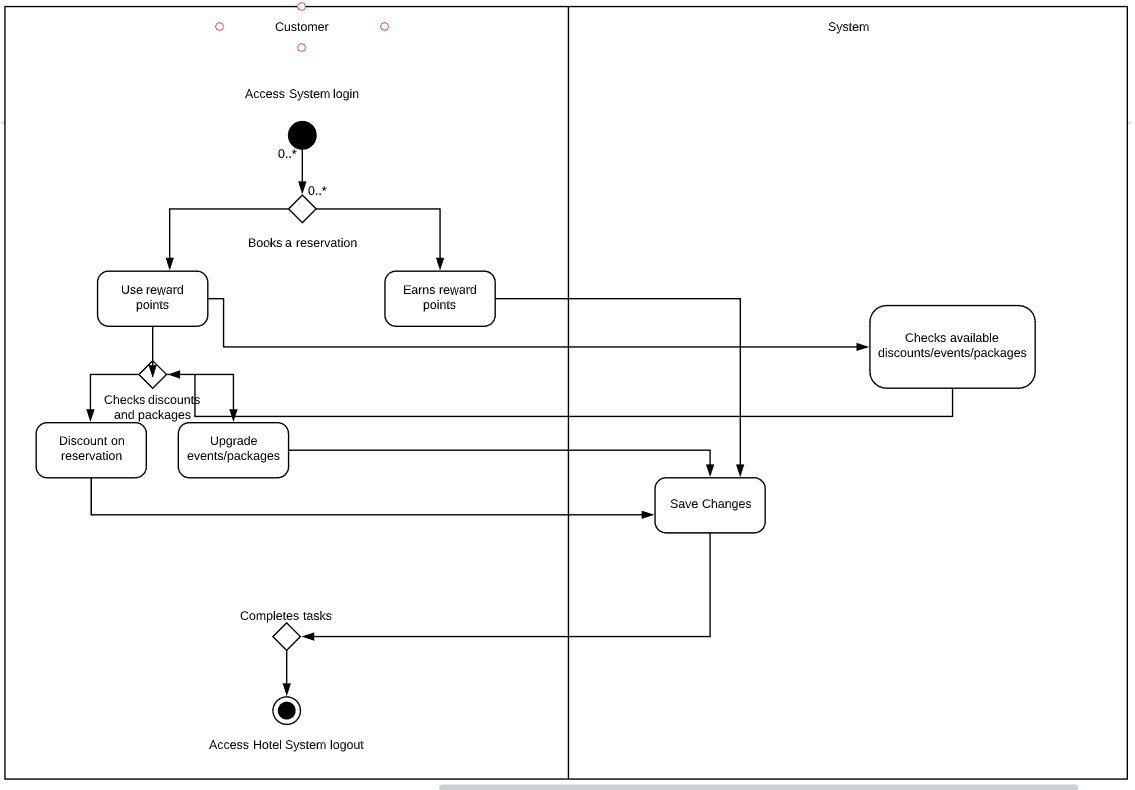
## **Rewards are applied to reservation if used.**

## **The cost of the reservation is lowered or reservations are upgraded through events/packages.**

* 1. **Successful viewing of history of points**

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

Draw diagram(s) that cover ALL main and alternate 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!

## **Customer Manages Rewards**

* + 1. Customers log into their hotel account.
    2. Customer clicks ‘View and Manage Rewards”
    3. Customer creates a reservation.
    4. Customers can use previously earned rewards points for discounts or events/packages.
    5. Customers earn rewards points for recently booked reservations.

## 

# **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!

**N/A**

# **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 | |
| Peter Schubert | Team Lead/ Software Developer | PS | 6/27/20 | |
| Jeffrey Fishman | Software Developer | JF | 6/27/20 | |
| Monier Abdullah | Software Developer | MA | 6/26/2020 | |
| Khalid Saeed | Software Developer | KS | 6/27/20 | |