|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. Use Case Überblick *(Arbeitspapier)* | | | | | | |
| Kriterien Use Case (diese Zeile kann nach der Erstellung gelöscht werden)Ein Anwendungsfall spezifiziert Aktionsfolgen (Szenarien) einschliesslich Alternativ- und Ausnahmesequenzen, die ein System oder eine Systemkomponente bei der Interaktion mit externen Objekten ausführt, um einen Mehrwert zu erbringen.Use Case beinhaltet: - Kontextinformatioenen, Ziele Vorbedingungen, Nachbedingungen- Hauptszenario; Ein Hauptszenario ist ein Szenario, das die Interaktionsfolge dokumentiert, die normalerweise ausgeführt wird, um eines oder mehrere mit dem Szenario assozierte Ziele zu erfüllen- Alternativszenario; Ein Alternativszenario ist ein Szenario, das zu einem Hauptszenario eine alternative Interaktionsfolge definiert. Eine Alternativszenario erfüllt die gleiche Ziele wie das zugehörige Hauptszenario- Ausnahmeszenario Ein Ausnahmeszenario ist ein Szenario, das eine Interaktionsfolge definiert, die ausgeführt wird, wenn in einem anderen Szenario (Haupt-, Alternativ- oder anderen Ausnahmeszenario) ein Ereignis eintritt, das die Erfüllung einer oder mehrerer mit dem Szenario assozierter Ziele verhindert | | | | | | |
| **Name** | Buchungen löschen | | | **ID** | | 201.010 |
| **Owner** | Remo Herren | | | **Rev-Number / Saved Date / Last Saved by** | | 1.2 / / |
| Storie / Notizen Diese Zeile dient dazu, dass Vorarbeiten von den Grobbeschrieben übernommen werden und kann nachträglich gelöscht werden. | | | | | | |
|  | |  | | | | |
| **Aktoren** | | Administratoren/Nutzer mit entsprechenden Berechtigungen | | | | |
| **Ziel** | | 1. Das checkouten, annullieren und löschen von Buchungen | | | | |
| **Vorbedingung** | | Reservationen sind vorhanden (in Listen, Plan oder Kalender) | | | | |
| **Hauptszenario** | | 1. Nutzer kann eine bestehende, aktive Buchung checkouten 2. Nutzer kann eine bestehende Buchung annullieren 3. Nutzer kann eine bestehende Buchung löschen | | | | |
| **Alternativszenario** | | Keines | | | | |
| **Ausnahmeszenario** | | Keines | | | | |
| **Fehlerbehandlung** | | Keines | | | | |
| **Regeln + Bedingungen** | | * Funktionsrechte müssen implementiert werden:   + Darf eigene Buchungen checkouten   + Darf Buchungen von Dritten checkouten   + Darf Buchungen von Dritten annullieren   + Darf eigene Buchungen löschen   + Darf Buchungen von Dritten löschen   + Darf eigene Buchungen direkt Checkouten   + Darf Buchungen von Dritten direkt checkouten   *Eigene Buchungen können immer annulliert werden.*  **Achtung:** Datenrechte müssen verifiziert werden. | | | | |
| **Qualitätsanforderungen** | | Keine | | | | |
| **Nachbedingung** | | Keine | | | | |
| **Beziehungen (UC)** | | Enthalten | Erweitert | | Vererbung | |
|  |  | |  | |

# Use Case Details



|  |  |
| --- | --- |
| **Schritt 1 -** Nutzer kann eine bestehende, aktive Buchung checkouten | |
| **Controls** | **Regeln** |
| **Inhaltsbereich**   * Button: „Checkout“ * Button: „Direkt Checkout“ | * Funktionsrechte müssen implementiert werden:   + Darf eigene Buchungen checkouten   + Darf Buchungen von Dritten checkouten   + Darf eigene Buchungen direkt Checkouten   + Darf Buchungen von Dritten direkt checkouten * Der Checkout-Button erscheint nur bei bestehenden, aktiven Buchungen und generell auch nur dann, wenn das entsprechende Funktionsrecht gegeben ist. * Der Checkout Button erscheint auch in der Nachlaufzeit einer Reservation * Wird der Checkout-Button geklickt, wird die aktive (laufende) Buchung auf den Zeitpunkt, an welchem der Button betätigt wurde, sekundengenau terminiert. * Es fallen nur die Reservations-Kosten gemäss Kostenmodell an. * In den Listen- und Kalender-Ansichten ist die Buchung bis zum terminierten End-Datum sichtbar. * Nach dem Checkout, wird die Buchung immer noch in den Reports (Auswertungen & Statistiken) geführt. * **Änderungs**notifikation an Ersteller, Organisator und allenfalls Dienstleister. Sofern die entsprechenden Anspruchgruppen die Notifikationen aktiviert haben.   **Spezialfall Direkt Checkout**   * Dieser Button wird nur angezeigt (Liste, View und Action PopUp) bei entsprechendem Funktionsrecht UND wenn auf der Reservation eine Nachlaufzeit vorhanden ist. * Direkt Checkout klick terminiert auch die Nachlaufzeit (sofort und sekundengenau). |
| Kein Bild | |

|  |  |
| --- | --- |
| **Schritt 2 -** Nutzer kann eine bestehende Buchung annullieren | |
| **Controls** | **Regeln** |
| **Inhaltsbereich**   * Button: „Annullieren“ | * Funktionsrechte müssen implementiert werden:   + Darf Buchungen von Dritten annullieren   + Info: eigene Buchungen können immer annulliert werden * Der Annullieren-Button erscheint standardmässig nur bei bestehenden aktiven und/oder zukünftigen eigenen Buchungen, sowie mit entsprechendem Funktionsrecht auch bei aktiven und/oder zukünftigen Buchungen für Dritte. * Wird der Annullieren-Button geklickt, erscheint eine Meldung, ob man wirklich annullieren will **(WICHTIG: diese zeigt auch die bevorstehenden Annullierungskosten an! – Meldung: Wollen Sie die Buchung wirklich annullieren? Die Annullierungskosten betragen CHF 10.--.),** bei Klick auf „Ja“, wird die aktive (laufende) oder zukünftige Buchung auf den Zeitpunkt, an welchem der Button betätigt wurde, annulliert. * In den Listen- und Kalender-Ansichten ist die Buchung nicht mehr sichtbar. * Nach der Annullation, wird die Buchung immer noch in den Reports (Auswertungen & Statistiken) geführt. * Verhalten bei einer aktiven Buchung:   + Es fallen nur die Reservations-Kosten gemäss Kostenmodell an. * Verhalten bei einer zukünftigen Buchung:   + Es fallen sowohl die Reservations- wie auch die Annullations-Kosten gemäss Kostenmodell an. * **Lösch**notifikation an Ersteller, Organisator und allenfalls Dienstleister. Sofern die entsprechenden Anspruchgruppen die Notifikationen aktiviert haben. * **Serieninformation**   Vergangenheit und laufende Buchung bleibt bestehen. Künftige Buchungen werden gelöscht. Annullationsmodell muss „greifen“. Bestätigungsmeldung an Organisator/Ersteller (1 E-Mail für die ganze Serie) |
| Kein Bild | |

|  |  |
| --- | --- |
| Schritt 3 - Nutzer kann eine bestehende Buchung löschen | |
| **Controls** | **Regeln** |
| **Inhaltsbereich**   * Button: Löschen | * Funktionsrechte müssen implementiert werden:   + Darf eigene Buchungen löschen   + Darf Buchungen von Dritten löschen * Der Löschen-Button erscheint nur bei bestehenden, aktiven und/oder zukünftigen, sofern das entsprechende Funktionsrecht gegeben ist. * Wird der Löschen-Button geklickt, wird die aktive (laufende) oder zukünftige Buchung auf den Zeitpunkt, an welchem der Button betätigt wurde, komplett aus dem System gelöscht. * Es fallen keine Reservation-s und auch keine Annullierungs-Kosten an. * In den Listen- und Kalender-Ansichten ist die Buchung nicht mehr sichtbar. * Nach dem Löschen, wird die Buchung auch nicht mehr in den Reports (Auswertungen & Statistiken) geführt. * **Lösch**notifikation an Ersteller, Organisator und allenfalls Dienstleister. Sofern die entsprechenden Anspruchgruppen die Notifikationen aktiviert haben. * **Serieninformation**   Alle Buchungen werden gelöscht, auch Vergangenheit. Bestätigungsmeldung an Organisator/Ersteller (1 E-Mail für die ganze Serie) |
| Kein Bild | |

# Anhang:

**Änderungsverlauf**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Ausführende Stelle | Bemerkungen |
| 1.0 | 14.04.2009 | Remo Herren | Erstellt |
| 1.1 | 23.04.2009 | Tim Bänziger | Review |
| 1.2 | 08.05.2009 | Tim Bänziger | Notifikationen eingebaut |
| 1.3 | 20.04.2010 | Tim Bänziger | Direkt Checkout eingebaut |