

# Kontrakte

## 1. Model

### Administration

1. addBooking
2. removeBooking
3. clean
4. AdHocDistribution.distribute
5. StandardDistribution.distribute
6. Timer.triggerBookingDistribution
7. run
8. filter

**Operation:** addBooking(Booking: booking)

**Beschreibung:** Hinzufügen einer Buchung zur Buchungsliste der Schedule.

**Vorbedingung:** Schedule und validierte Buchung existieren. (Sind nicht null)

**Nachbedingung:** Die Liste bookings von Schedule wurde um die übergebene valide Buchung erweitert.

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** Schedule enthält die Buchung bereits.

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** removeBooking(Booking: booking)

**Beschreibung:** Entfernt eine Buchung aus der Buchungsliste einer Schedule.

**Vorbedingung:** Schedule und zu entfernende Buchung existieren.

**Nachbedingung:** Buchung wurde aus der Schedule entfernt.

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** Schedule enthält die Buchung nicht.

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** clean(DateTime: now)

**Beschreibung:** Entferne alle abgelaufenen Buchungen aus der Buchungsliste der Schedule.

**Vorbedingung:** Schedule existiert.

**Nachbedingung:** Schedule enthält keine abgelaufenen Buchungen mehr, d. h. für jede Buchung buchung gilt: buchung.getEndDate() >= now.

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** StandardDistribution.distribute(Booking[1..\*]: bookings, Schedule: schedule)

**Beschreibung:** Verteile alle vorhandenen Buchungen in die Schedule.

**Vorbedingung:** Schedule existiert. Nichtleere Liste zu verteilender und valider Buchungen auf der entsprechenden Schedule existiert.

**Nachbedingung:** Übergebene Buchungen, wurden nun in die Buchungsliste der entsprechenden Schedule verteilt. Attribut "station" der Buchungen wurde bei der Zuweisung gesetzt. Der Startzeitpunkt der Buchungen wurde auf den Beginn des Ladevorgangs gesetzt und der Endzeitpunkt entsprechend auf das Ende.

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** Eine der Buchungen ist bereits in der Schedule enthalten. (Operation addBooking(Booking: booking) gibt den Wahrheitswert false zurück.)

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** AdHocDistribution.distribute(Booking[1..\*]: bookings, Schedule: schedule)

**Beschreibung:** Verteile alle AdHoc-Buchungen in die Schedule.

**Vorbedingung:** Schedule existiert. Eine valide AdHoc Buchung auf dem zur Schedule gehörenden Standort wurde durchgeführt.

**Nachbedingung:** Die AdHoc Buchung wurde der Schedule hinzugefügt. Attribut "station" der Buchung wurde gesetzt. Der Startzeitpunkt der Buchungen wurde auf den Beginn des Ladevorgangs gesetzt und der Endzeitpunkt auf das Ende des Laden gesetzt

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** Die Buchung ist bereits in der Schedule enthalten. (Operation addBooking(Booking: booking) gibt den Wahrheitswert false zurück.)

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** Timer.triggerBookingDistribution()

**Beschreibung:** Aktiviere den Distributor regelmäßig zu einem vorgegebenen Zeitpunkt um normale Buchungen zu verteilen.

**Vorbedingung:** Distributor und zugehörige Schedule existieren. Es gibt zu verteilende Buchungen.

**Nachbedingung:** Die Methode run() wurde auf dem Distributor zum vorgegebenen Zeitpunkt aufgerufen.

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** Es gibt keine Buchungen zum Verteilen. Die Kapazitäten sind erschöpft.

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** AdHoc.triggerBookingDistribution()

**Beschreibung:** Aktiviere den Distributor um AdHoc Buchungen zu verteilen. Setze die Verteilungsstrategie des Distributors auf AdHoc.

**Vorbedingung:** Distributor und zugehörige Schedule existieren. Es wurde eine valide AdHoc Buchung aufgegeben.

**Nachbedingung:** Die Methode run() wurde auf dem Distributor zum vorgegebenen Zeitpunkt aufgerufen.

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** Es gibt keine AdHoc-Buchung zum Verteilen. Die Kapazitäten sind erschöpft.

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** run()

**Beschreibung:** Führe die Buchungsverteilung durch. Sorgt für die Benachrichtigung per E-Mail.

**Vorbedingung:** Es wurde eine Verteilungsstrategie gesetzt.

**Nachbedingung:** Es gibt im Buchungscache keine Buchungen mehr, die das Filterkriterium vom zugehörigen Filter erfüllen. Diese sind in dem Zugehörigen schedule. Der Notification Manager wurde für jede Booking informiert.

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---

**Operation:** filter(Booking[1..\*]: bookings, DateTime: date)

**Beschreibung:** Filter die Buchungsliste nach einem bestimmten Merkmal.

**Vorbedingung:** Die Buchungsliste ist nicht leer.

**Nachbedingung:** Es existiert eine Buchungsliste mit den gefilterten Buchungen.

**Ergebnisse:** Eine Liste and Buchungen, die das Filterkriterium erfüllen.

**Ausnahmen:** Kein Objekt erfüllt das Filterkriterium.

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** O(n)

## Simulation

1. generateDateTimeValues
2. addVehicle
3. deleteVehicle
4. addRushhour
5. deleteRushhour
6. updateWorkload
7. createBooking toDO
8. triggerBookinDistribution to DO

9. Simulation.run
10. calculataLocationworkload
11. calculateSation
12. startGenerator

**Operation:** generateDateTimeValues(DateTime: start, DateTime: end, int: bookings)

**Beschreibung:** generiert im bereich von start zu end bookings Zeitpunkte die normalverteilt sind.

**Vorbedingung:** start existiert, end existiert und ist größer als start, bookings existiert und ist größer 0

**Nachbedingung:** ---

**Ergebnisse:** Elne Liste von DateTimes zwischen start und end (Normalverteilt)

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---

**Operation:** addVehicle(Vehicle: vehicle)

**Beschreibung:** fügt vehicle der Liste der enthaltenen Vehicle hinzu

**Vorbedingung:** vehicle existiert und ist valide.

**Nachbedingung:** vehicle wurde der Liste der Vehicle hinzugefügt

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---

**Operation:** deleteVehicle(Vehicle: vehicle)

**Beschreibung:** Entfernt vehicle aus der Liste der enthaltenen Vehicle.

**Vorbedingung:** vehicle existiert und ist valide.

**Nachbedingung:** vehicle wurde der Liste der Vehicle entfernt.

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** vehicle ist nicht in der Liste der Vehicle.

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---

**Operation:** addRushhour(Rushhour: rushhour)

**Beschreibung:** Fügt rushhour der Liste der enthaltenen Rushhour hinzu.

**Vorbedingung:** rushhour existiert und ist valide.

**Nachbedingung:** rushhour wurde der Liste der Rushhour hinzugefügt.

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** rushhour ist bereits in der Liste der Rushhour enthalten.

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---

**Operation:** deleteRushhour(Rushhour: rushhour)

**Beschreibung:** Entfernt rushhour aus der Liste der enthaltenen Rushhour.

**Vorbedingung:** rushhour existiert und ist valide.

**Nachbedingung:** rushhour wurde der Liste der rushhour entfernt.

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** rushhour ist nicht in der Liste der Rushhour .

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---

**Operation:** updateWorkload(double: location, double[1..\*]: station)

**Beschreibung:** aktualisiert den Workload, erweitert stationWorkload und locationWorkload um jeweils einen eintrag

**Vorbedingung:** location existiert und ist valide. station ist valide und existiert.

**Nachbedingung:** location ist der neue letzte eintrag von locationWorkload station ist der neue letzte eintrag von stationWorkload

**Ergebnisse:** Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---



**Operation:** createBookings(Scenario: scenario)

**Beschreibung:** Erstellt eine Booking mit den übergebenen Parameter.

**Vorbedingung:** die Übergabeparameter ergeben eine Valide Booking

**Nachbedingung:** Eine neue Booking ist im cache enthalten

**Ergebnisse:**

**Ausnahmen:**

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---

**Operation:** triggerBookingDistribution(DateTime: now)

**Beschreibung:**

**Vorbedingung:**

**Nachbedingung:**

**Ergebnisse:**

**Ausnahmen:**

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---

**Operation:** Simulator.run()

**Beschreibung:** ruft generateDateTimeValues auf und übergibt bookings, end, start

**Vorbedingung:** ---

**Nachbedingung:** ---

**Ergebnisse:** gibt Rückgabewert von generateDateTimeValue zurück.

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---

**Operation:**calculateLocationWorkload(now: DateTime)

**Beschreibung:** berechnet den Workload der Location zum aktuellen timestep now innerhalb der Simulation

**Vorbedingung:** now ist existiert und ist ein Zeitpunkt innerhalb des Szenarios

**Nachbedingung:** ---

**Ergebnisse:** gibt die Auslastung der Location zurück

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** ---

**Operation:**calculateStationWorkload(now: DateTime)

**Beschreibung:** berechnet den Workload der Stations der Location zum aktuellen timestep now innerhalb der Simulation

**Vorbedingung:** now ist existiert und ist ein Zeitpunkt innerhalb des Szenarios

**Nachbedingung:** ---

**Ergebnisse:** gibt die Auslastung der Stations zurück

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** Systemoperation

**Querverweise:** ---

**Anmerkung:** die Reihenfolge innerhalb des rückgabe arrays entspricht der Reihenfolge wenn man Location.zones[0].stations[0] über Location.zones[0].stations[n] hinzu Location.zones[n].stations[n] durchläuft.

## History

### Infrastructure

1. addZone
2. deleteZone
3. addStation
4. deleteStation
5. addPlug
6. deletePlug
7. CompareTo

<b>Operation:</b>	addZone( Zone: zone) : bool
<b>Beschreibung:</b>	Hinzufügen einer Zone zu Zonenliste einer Location.
<b>Vorbedingung:</b>	Eine hinzuzufügende valide Zone existiert .
<b>Nachbedingung:</b>	Die Zonenliste wurde um Zone erweitert.
<b>Ergebnisse:</b>	Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht.
<b>Ausnahmen:</b>	Zone ist bereits enthalten.
<b>Ausgaben:</b>	---
<b>Typ:</b>	Systemoperation
<b>Querverweise:</b>	---
<b>Anmerkung:</b>	---

<b>Operation:</b> <b>Beschreibung:</b> <b>Vorbedingung:</b> <b>Nachbedingung:</b> <b>Ergebnisse:</b>  <b>Ausnahmen:</b> <b>Ausgaben:</b> <b>Typ:</b> <b>Querverweise:</b> <b>Anmerkung:</b>	deleteZone(Zone: zone): bool Entfernen einer Zone aus der Zonenliste. Zone existiert. Zone ist aus der Zonenliste entfernt. Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht. Zone ist nicht in der Zonenliste enthalten. --- Systemoperation --- ---
<b>Operation:</b> <b>Beschreibung:</b> <b>Vorbedingung:</b> <b>Nachbedingung:</b> <b>Ergebnisse:</b>  <b>Ausnahmen:</b>  <b>Ausgaben:</b> <b>Typ:</b> <b>Querverweise:</b> <b>Anmerkung:</b>	addStation(Station:station) : bool Hinzufügen von Station zur Stationenliste. Valide Station existiert. Station ist in der Stationenliste enthalten. Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht. Station ist bereits enthalten. --- Systemoperation --- ---

<b>Operation:</b> <b>Beschreibung:</b> <b>Vorbedingung:</b> <b>Nachbedingung:</b> <b>Ergebnisse:</b>  <b>Ausnahmen:</b> <b>Ausgaben:</b> <b>Typ:</b> <b>Querverweise:</b> <b>Anmerkung:</b>	deleteStation(Station:station): bool Entfernen einer Station aus der Stationenliste”. Valide Station existiert. Station wurde aus der Stationenliste entfernt. Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht. Station ist nicht in der Stationenliste enthalten. --- Systemoperation --- ---
<b>Operation:</b> <b>Beschreibung:</b> <b>Vorbedingung:</b> <b>Nachbedingung:</b> <b>Ergebnisse:</b>  <b>Ausnahmen:</b> <b>Ausgaben:</b> <b>Typ:</b> <b>Querverweise:</b> <b>Anmerkung:</b>	addPlug(Plug: plug): bool Hinzufügen von Plug zu der Plugliste. Valider Plug existiert. Plug wurde zu der Plugliste hinzugefügt. Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht. Plug ist bereits in der Plugliste enthalten. --- Systemoperation --- ---
<b>Operation:</b> <b>Beschreibung:</b> <b>Vorbedingung:</b> <b>Nachbedingung:</b> <b>Ergebnisse:</b>  <b>Ausnahmen:</b> <b>Ausgaben:</b> <b>Typ:</b> <b>Querverweise:</b> <b>Anmerkung:</b>	deletePlug(Plug: plug) : bool Entfernen von Plug der Liste “plugins”. Plug existiert. Plug wurde aus der Liste “plugins” entfernt. Wahrheitswert entsprechend ob die Operation erfolgreich war oder nicht. Plug ist nicht in der Liste “plugins” enthalten. --- Systemoperation --- ---

<b>Operation:</b>	compareTo(object obj) : bool
<b>Beschreibung:</b>	vergleicht das obj mit der Zone.
<b>Vorbedingung:</b>	obj ist eine Zone
<b>Nachbedingung:</b>	---
<b>Ergebnisse:</b>	Wahrheitswert entsprechend ob der char bei site gleich war
<b>Ausnahmen:</b>	---
<b>Ausgaben:</b>	---
<b>Typ:</b>	Systemoperation
<b>Querverweise:</b>	---
<b>Anmerkung:</b>	---

## Communication

1. update()
2. generateMessageAccepted
3. generateMessageDeclined
4. generateMessageCheckIn
5. generateMessageCheckOut
6. sendMessage

<b>Operation:</b> update( Booking :booking, string: eventName)
<b>Beschreibung:</b> entscheidet mithilfe des eventNames welche Nachricht an die E-mail Adresse bookings.user geschickt werden soll und sorgt dafür das diese verschickt wird
<b>Vorbedingung:</b> booking ist valide, eventName ist ein eventName und existiert
<b>Nachbedingung:</b> der Messenger verschickt eine E-Mail
<b>Ergebnisse:</b> ---
<b>Ausnahmen:</b> booking.User ist nicht gesetzt (es passiert einfach nichts)
<b>Ausgaben:</b> ---
<b>Typ:</b> ---
<b>Querverweise:</b> ---
<b>Anmerkungen:</b> ---

**Operation:** generateMessageAccepted( Booking :booking)

**Beschreibung:**generiert mithilfe von commands und den details aus booking die Nachricht für den User

**Vorbedingung:** booking ist valide und existiert

**Nachbedingung:** ---

**Ergebnisse:** Nachricht für den User

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** generateMessageDeclined( Booking :booking)

**Beschreibung:**generiert mithilfe von commands und den details aus booking die Nachricht für den User

**Vorbedingung:** booking ist valide und existiert

**Nachbedingung:** ---

**Ergebnisse:** Nachricht für den User

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** generateMessageCheckIn( Booking :booking)

**Beschreibung:**generiert mithilfe von commands und den details aus booking die Nachricht für den User

**Vorbedingung:** booking ist valide und existiert

**Nachbedingung:** ---

**Ergebnisse:** Nachricht für den User

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** generateMessageCheckOut( Booking :booking)

**Beschreibung:**generiert mithilfe von commands und den details aus booking die Nachricht für den User

**Vorbedingung:** booking ist valide und existiert

**Nachbedingung:** ---

**Ergebnisse:** Nachricht für den User

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---



**Operation:** sendMessage(message: string, user: string)

**Beschreibung:** sendet die Nachricht message an die E-Mail user

**Vorbedingung:** message existiert, user existiert und ist eine E-Mail

**Nachbedingung:** Der richtige Nutzer erhält eine E-Mail.

**Ergebnisse:** ---

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

## User

### 1. getUserPriority

**Operation:** getUserTyp(String: email)

**Beschreibung:** Ließt aus einem Textdokument die Priorität des Nutzers. Ist diese nicht enthalten wird die Rolle Gast zurück gegeben.

**Vorbedingung:** Email existiert

**Nachbedingung:** ---

**Ergebnisse:** Rolle des Users

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

## 2. Controller

### HomeController

1. Index
2. Imprint
3. Help
4. Login

**Operation:**Index()

**Beschreibung:** gibt die Home Seite zurück.

**Vorbedingung:** ---

**Nachbedingung:** Der User sieht die Home Seite

**Ergebnisse:** Die Seite Index wird zurückgegeben

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:**Imprint()

**Beschreibung:** gibt die Impressum Seite zurück.

**Vorbedingung:** ---

**Nachbedingung:** Der User sieht die Impressum Seite

**Ergebnisse:** Die Seite Imprint wird zurückgegeben

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:**Help()

**Beschreibung:** gibt die Help Seite zurück.

**Vorbedingung:** ---

**Nachbedingung:** Der User sieht die Help Seite

**Ergebnisse:** Die Seite Help wird zurückgegeben

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:** Login(string: email)

**Beschreibung:** Login vorgang des Users

**Vorbedingung:** email ist nicht leer, email ist eine valide email

**Nachbedingung:** Rolle wurde als SessionVariable gesetzt, der Nutzer sieht die Admin.Dashboard Seite sollte er die Rolle PLANER haben, sonst sieht er die Booking.Index Seite.

**Ergebnisse:** ---

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

## BookingController

Index

Create

Post

Edit

Delete

ToggleCheck

**Operation:**Index()

**Beschreibung:** Erstellt ein DashboardViewModel und gibt die Booking.Index Seite zurück, sollte seine Rolle PLANER sein landet er auf der Admin.Dashboard Seite.

**Vorbedingung:**

**Nachbedingung:** Der User sieht die Booking.Index Seite auf der seine Buchungen angezeigt werden

**Ergebnisse:** Die Seite Booking.Index wird zurückgegeben

**Ausnahmen:** Der User hat sich noch nicht eingeloggt

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:**Create()

**Beschreibung:** Erstellt die BookingCreateViewModel und gibt die Booking.Create Seite zurück

**Vorbedingung:** Der User ist eingeloggt, und ist berechtigt diese Seite zu sehen

**Nachbedingung:** Der User sieht die Booking.Create Seite auf der er eine neue Buchung erstellen kann.

**Ergebnisse:** Die Seite Booking.Create wird zurückgegeben

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:**Post(Booking: booking)

**Beschreibung:** gibt die Booking.Index Seite zurück und fügt die booking in den Cache hinzu

**Vorbedingung:** Der User ist eingeloggt, booking existiert.

**Nachbedingung:** Der User sieht die Booking.Index Seite auf der er seine Buchungen sehen kann, auf der auch die neue Buchung zu sehen ist.

**Ergebnisse:** Die Seite Booking.Index wird zurückgegeben

**Ausnahmen:** booking ist nicht valide.

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:**Edit(Booking: booking)

**Beschreibung:** gibt die Booking.Create Seite zurück

**Vorbedingung:** Der User ist eingeloggt, booking existiert, ist im cache und ist valide

**Nachbedingung:** Der User sieht die Booking.Create Seite auf der er seine Buchung bearbeiten kann. Das Formular ist ausgefüllt mit den Werten aus booking. booking ist nicht mehr im Cache

**Ergebnisse:** Die Seite Booking.Create wird zurückgegeben

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:**Delete(Booking: booking)

**Beschreibung:** gibt die Booking.Index Seite zurück.

**Vorbedingung:** Der User ist eingeloggt, booking existiert, ist im cache und ist valide

**Nachbedingung:** Der User sieht die Booking.Index auf der er seine Buchungen sehen kann, booking ist nicht mehr zu sehen. Booking ist nicht mehr im Cache

**Ergebnisse:** Die Seite Booking.Create wird zurückgegeben

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

**Operation:**ToggleCheck(Booking: booking)

**Beschreibung:** Regelt den Check-In, Check-Out Vorgang.

**Vorbedingung:** Der User ist eingeloggt, booking existiert, es ist der richtige Zeitpunkt für diese Aktion.

**Nachbedingung:** Der User sieht die Booking.Index Seite auf der er seine Buchungen sehen kann. booking.active ist true sollte es vorher false gewesen sein und false sollte es vorher true gewesen sein.

**Ergebnisse:** Die Seite Booking.Index wird zurückgegeben

**Ausnahmen:** ---

**Ausgaben:** ---

**Typ:** ---

**Querverweise:** ---

**Anmerkungen:** ---

### **3. ViewModel**

### **4. Persistence**