# User

## attributes

|  |  |
| --- | --- |
| **Attribute name** | **Notes** |
| Id \* | Number, read only, auto-increment |
| Name \* | String, min 3 max 30 |
| Phone \* | String, matches Egypt phone numbers format |
| Profile image | String, image link |
| Password \* | String, min 8 max 16 |
| Wishlist | [room-ref] |

## Business Operations

### Business Operation: **create user**

#### Validation Rules

* Phone must be unique

#### ACL Rules

* Anonymous

# Available services

## attributes

|  |  |
| --- | --- |
| **Attribute name** | **Notes** |
| Id \* | Number, read only, auto-increment |
| Name \* | String, min 3 max 30 |

## Business Operations

### Business Operation: **create/ update/ delete**

#### Overview

* Example:
* Available-Service-TV
* Available-Service -Wi Fi
* Available-Service -Kitchen
* Available-Service -Laptop
* Available-Service -Iron

#### Validation Rules

#### ACL Rules

* Admin

# Room

## attributes

|  |  |
| --- | --- |
| **Attribute name** | **Notes** |
| Id \* | Number, read only, auto-increment |
| User id \* | Number Read only, User ref |
| Name \* | String, min 3 max 30 |
| Images \* | Min 1 max 5 |
| Location \* | map location |
| Night price \* | Number, min 0 |
| Capacity \* | Number, min 1 |
| Available Services | [available-service ref] |
| Notes | String |
| Active | Boolean, default true |

## Business Operations

### Business Operation: **create room**

#### ACL Rules

* User

### Business Operation: **search rooms**

#### Search Criteria

* Available-Services
* Night price range (from/to)
* Booking duration (search duration doesn’t intersect with orders with status: PENDING, ACCEPTED)
* Capacity
* Location (default: user’s location)
* Distance

#### Sort criteria

* By distance (default sorting: the distance between the room location and the search-location)
* By Night Price (ascending / descending)

### Business Operation: **get my rooms list**

#### Validation Rules

* User id

#### ACL Rules

* User

### Business Operation: **add to/ remove from Wishlist**

#### Overview

* The user can review rooms and add them to/ remove from his Wishlist

#### Validation Rules

* The user can’t add his rooms to his Wishlist

#### ACL Rules

* User

### Business Operation: **update room**

#### ACL Rules

* User

### Business Operation: **delete room**

#### Overview

* When deleting a room, delete if from wish-lists if existed.

#### ACL Rules

* User

# Order

## attributes

|  |  |
| --- | --- |
| **Attribute name** | **Notes** |
| Id \* | Number, read only, auto-increment |
| User id\* | Number, read only, User ref |
| Room \* | Number, Room ref |
| Booking Period \* | [from: number, to: number]  Default hour: **12:00PM**  Ex: the user sets: **16/11/2021** the system saves it as **16/11/2021 12:00PM** |
| Notes | String |
| Order price \* | Number, calculated (booking period in days \* room-night-price) |
| Paid | Boolean, default: false |
| Status \* | String, default PENDING, options: [PENDING, ACCEPTED, REFUSED, COMPLETED, CANCELLED, EXPIRED] |
| Cancel notes | String |

## Business Operations

### Business Operation: **create order**

#### Overview

* a user can book a specific room for a specific period
* the user determines the booking period start and end days (and the system sets a default time 12:00PM)
* booking-period-start date must be greater than current-date by 1 day at least
* a notification will be sent to the room-owner to inform him with the created-order (“new booking order created on room {id}”)

#### Validation Rules

* The user can’t create order on his rooms
* Booking period must be in the future (greater than current-date by 1 day at least)
* The user can’t create an order with a period intersected with an order with status (pending, accepted)

#### ACL Rules

* User

### Business Operation: **edit order**

#### Validation Rules

* Status = PENDING

#### ACL Rules

* Order-owner

### Business Operation: **expire pending orders**

#### Overview

* If the room owner didn’t take any action in 3 hours before booking-period\_start-date, the pending order will be expired
* Status will be updated to be EXPIRED
* A notification will be sent to the room-owner to inform him with the expired-order (“order {id} on room {id} has been expired”)
* A notification will be sent to the order-owner to inform him with the expired-order (“order {id} on room {id} has been expired”)

#### Validation Rules

* status = PENDING

#### ACL Rules

* automatically by the system

### Business Operation: **accept order**

#### Overview

* The room owner can accept pending orders on his rooms
* A notification will be sent to the order-owner to inform him that his order bas been accepted and he has 2hours to pay for that order (“order {id} on room {id} has been accepted”)

#### Validation Rules

* status = PENDING

#### ACL Rules

* room owner

### Business Operation: **refuse order**

#### Overview

* The room owner can refuse pending orders on his rooms
* The status will be updated to be REFUSED
* A notification will be sent to the order-owner to inform him that his order bas been refused (“order {id} on room {id} has been refused”).

#### Validation Rules

* status = PENDING

#### ACL Rules

* room owner

### Business Operation: **expire accepted orders**

#### Overview

* if the order owner didn’t pay for an accepted order within 2hours after his order being accepted by the room-owner, his order will be expired.
* Status will be updated to be EXPIRED
* A notification will be sent to the room-owner to inform him with the expired-order (“order {id} on room {id} has been expired”)
* A notification will be sent to the order-owner to inform him with the expired-order (“order {id} on room {id} has been expired”)

#### Validation Rules

* status = ACCEPTED
* paid = false
* 2 hours passed after being accepted without being paid

#### ACL Rules

* automatically by the system

### Business Operation: **complete accepted orders**

#### Overview

* when the booking-period\_end-date reached, the system updates the status of the accepted and paid orders to be COMPLETED
* a notification will be sent to both room-owner and order-owner (“order {id} on room {id} has been completed”)

#### Validation Rules

* status = ACCEPTED
* paid = true

#### ACL Rules

* automatically by the system

### Business Operation: **pay for an order accepted orders**

#### Overview

* If an order being accepted, the order-owner must pay for that order within 2hours
* (paid) attribute will be updated to be (true)
* A notification will be sent to the room-owner to inform him with the payment (“order {id} on room {id} has been paid”)

#### Validation Rules

* status = ACCEPTED
* paid = false
* paying within 2hours after acceptance

#### ACL Rules

* order-owner

### Business Operation: **cancel pending orders**

#### Overview

* status will be updated to be CANCELLED
* a notification will be sent to the room owner
* (Cancel notes) will be updated to be (cancelled by the order-owner)

#### Validation Rules

* status = PENDING

#### ACL Rules

* order-owner

### Business Operation: **cancel accepted orders**

#### Overview

* status will be updated to be CANCELLED
* when cancelled by the room-owner:
* a notification will be sent to the order-owner
* (Cancel notes) will be updated to be (cancelled by the room-owner)
* when cancelled by the order-owner:
* a notification will be sent to the room-owner
* (Cancel notes) will be updated to be (cancelled by the order-owner)
* If (paid == true), a REFUND process will be implemented automatically.

#### Validation Rules

* status = ACCEPTED

#### ACL Rules

* order-owner
* room-owner

### Business Operation: **reminder with a room-booking**

#### Overview

* when reaching the start-date of the booking period, the system sends a notification to both the room-owner and the order-owner to notify them with the booking (“order {id} on room {id} has been started”)

#### Validation Rules

* status = ACCEPTED
* paid = true

#### ACL Rules

* automatically by the system

### Business Operation: **check room availability**

#### Model

* Room

#### Overview

* Return (true or the room is available, if there is no pending or accepted order intersected with the search period

#### Search criteria

* from date/ to date

### Business Operation: **get my orders**

#### Search criteria

* User id

#### Sort criteria

* Booking-period\_start-date (asc)

#### ACL Rules

* User

### Business Operation: **get orders received on my rooms**

#### Search criteria

* Room.user-id

#### Sort criteria

* Booking-period\_start-date (asc)

#### ACL Rules

* User

# Room (update)

### Business Operation: **update room**

#### Overview

* Updates on rooms assigned to orders (especially updating the night-price), affects only the subsequent orders.

#### ACL Rules

* User

### Business Operation: **delete room**

#### Overview

* Rooms assigned to orders can’t be deleted

#### ACL Rules

* User

### Business Operation: **activate / deactivate room**

#### Overview

#### ACL Rules

* User

# Chat module (general chat)

## Overview

* The app users can chat with each other