## RECEPTIONIST PARCEL APP

- Receptionist manages the Guests and their Parcels.
- These Guests can Check-In and Check-Out anytime.
- Guests have Parcels which the Receptionist Receives when Guests are Checked-In.
- Guests can Pick-up Parcel from the Receptionist.
- Receptionist Must not Receive Parcel for Checked-Out Guests.
- Receptionist hands over Unpicked Parcels at Guest Check-Out.
- Guests can Only Check-Out when All Parcels are Picked-Up.

#### Requirements (Specific):

- 1. Guests can Check-in with Name & Details.
- 2. Receptionist Checks if Guest is Checked-In.
- 3. Receives Parcel under (checked-in) Guest Name with Parcel Code.
- 4. Guests Pick-up Parcel at the Reception by Parcel Code.
- 5. Receptionist Checks List of Unpicked (Available) Parcels for Guest.
- 6. Guests can Check-out when All Parcels are Picked-up.

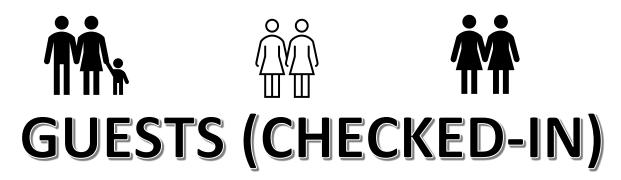
### Tasks & Technology Stack:

- 1. Use Java 8 & Spring Boot.
- 2. Interceptor Adaptor for Logging Handles.
- 3. Junit 5 Test Cases.
- 4. Restful JSON-API in Designed Microservices.
- 5. Design & Usability Document (PDF file).
- 6. Swagger File Documentation & Testing.

#### **Deliverables:**

- 1. Source Code in GitHub Public repository.
- 2. Maven Build Spring Boot Project with Executable Jar.
- 3. README file to run and access the REST JSON-API.

















PARCEL LIST (EACH GUEST)

# **MODELS & RELATIONS**

**Model Name: GUEST** 

### **Attributes:**

- 1. Long Auto Generated Guest Id.
- 2. ZonedDateTime checkinDate for Check-in Tracking.
- 3. Boolean ischeckedin for Guest Status.
- 4. String name Unique Guest Naming.
- 5. String details Guest Details (Expandable as Required).
- 6. ZonedDateTime checkoutDate for Guest Check-out Tracking.
- 7. Set<Parcel> Unique Group of parcels for each Guest.

## **Model Name: PARCEL**

### **Attributes:**

- 1. Long Auto Generated Parcel Id.
- 2. ZonedDateTime **receiveDate** for Parcel Tracking.
- 3. Boolean ispicked to notify if Parcel picked-up by Guest.
- 4. String **code** Unique Parcel Naming/Scanning.
- 5. ZonedDateTime pickedupDate for Parcel pick-up Tracking.

Relation: One to Many {Guest -> Parcel Set}

Joined as Owner of Parcel

Referenced Name of Guest

Cascading All Types with Orphan Removal.

# **API Design & DTO**

### **GUEST CONTROLLER (reception/guests)**

**GET Mapping (/)** findAllGuests

Input: No Parameters

Output: Guest List.

POST Mapping (/) createGuest

Input: Guest (Name & Details)

Output: Guest.

PUT Mapping (/) updateGuest HARD

Input: Guest (Changes with Id)

Output: None.

DELETE Mapping (/guestName) deleteGuest\_HARD

Input: String Guest Name

Output: None.

GET Mapping (/guestName) getGuestByName

Input: String Guest Name

Output: Guest.

Basic CRUD and Search/Delete by Guest Name.

#### PARCEL CONTROLLER (reception/parcels)

**GET Mapping (/)** findAllParcels

Input: No Parameters

Output: Parcel List

PUT Mapping (/) updateParcel\_HARD(Guest)

Input: Parcel (Changes with Id)

Output: None.

GET Mapping (/parcelCode) getParcelByCode

Input: String Parcel Code

Output: Parcel.

PUT Mapping (/parcelCode) parcelPickedByGuest

Input: String Parcel Code

Output: None.

DELETE Mapping (/parcelCode) deleteParcel\_HARD

Input: String Parcel Code

Output: None.

POST Mapping (/parcelCode/guestName) createParcel

Input: Parcel & String Guest Name

Output: Parcel.

Basic Read, Search by Parcel Code, Delete by Parcel Code.

Updating Parcel (ispicked) when Guest Picks Parcel.

Creating Parcel Only when Guest Available and Checked-In.

**Exception: GuestAlreadyCheckedOutException** 

#### CHECKOUT CONTROLLER (reception/checkout)

GET Mapping (/parcels/guestName) parcelsByGuestName

Input: String Guest Name

Output: Parcel List

PUT Mapping (/parcels/guestName) pickedAllParcelsByGuest

Input: String Guest Name

Output: None.

GET Mapping (/status/guestName) getGuestStatusByName

Input: String Guest Name

Output: CHECKED-IN / CHECKED-OUT

PUT Mapping (status/guestName) guestCheckout\_HARD

Input: String Guest Name

Output: None.

Receptionist can check Guest Status Anytime.

Receptionist can check Unpicked Parcels for Guest.

Guest can Pick all Parcels at Once.

Guest can Check-out when All Parcels Picked.

**Exception: UnpickedPaceIsException** 

# **SWAGGER DOCUMENTATION &**

# **TESTING API**

Swagger User Interface (/swagger-ui.html)

• Controllers:

Home Controller Swagger UI
Guest Controller Guest CRUD+
Parcel Controller Parcel CRUD+
Checkout Controller Checkout RU+

• Models:

Guest

**Parcel** 

• Data Patch and Testing:

src/main/resources data.sql