

# 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.



# HOTEL RECEPTIONIST



## GUESTS (CHECKED-IN)



## 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)

<b>GET Mapping (/)</b>	<b>findAllGuests</b>
<b>Input:</b>	<b>No Parameters</b>
<b>Output:</b>	<b>Guest List.</b>
<b>POST Mapping (/)</b>	<b>createGuest</b>
<b>Input:</b>	<b>Guest (Name &amp; Details)</b>
<b>Output:</b>	<b>Guest.</b>
<b>PUT Mapping (/)</b>	<b>updateGuest_HARD</b>
<b>Input:</b>	<b>Guest (Changes with Id)</b>
<b>Output:</b>	<b>None.</b>
<b>DELETE Mapping</b>	<b>(/guestName) deleteGuest_HARD</b>
<b>Input:</b>	<b>String Guest Name</b>
<b>Output:</b>	<b>None.</b>
<b>GET Mapping</b>	<b>(/guestName) getGuestByName</b>
<b>Input:</b>	<b>String Guest Name</b>
<b>Output:</b>	<b>Guest.</b>

**Basic CRUD and Search/Delete by Guest Name.**

## **PARCEL CONTROLLER (reception/parcels)**

<b>GET Mapping (/)</b>	<b>findAllParcels</b>
<b>Input:</b>	<b>No Parameters</b>
<b>Output:</b>	<b>Parcel List</b>
<b>PUT Mapping (/)</b>	<b>updateParcel_HARD(Guest)</b>
<b>Input:</b>	<b>Parcel (Changes with Id)</b>
<b>Output:</b>	<b>None.</b>
<b>GET Mapping</b>	<b>(/parcelCode) getParcelByCode</b>
<b>Input:</b>	<b>String Parcel Code</b>
<b>Output:</b>	<b>Parcel.</b>
<b>PUT Mapping</b>	<b>(/parcelCode) parcelPickedByGuest</b>
<b>Input:</b>	<b>String Parcel Code</b>
<b>Output:</b>	<b>None.</b>
<b>DELETE Mapping</b>	<b>(/parcelCode) deleteParcel_HARD</b>
<b>Input:</b>	<b>String Parcel Code</b>
<b>Output:</b>	<b>None.</b>
<b>POST Mapping</b>	<b>(/parcelCode/guestName) createParcel</b>
<b>Input:</b>	<b>Parcel &amp; String Guest Name</b>
<b>Output:</b>	<b>Parcel.</b>

**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: UnpickedPacelsException**

# 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