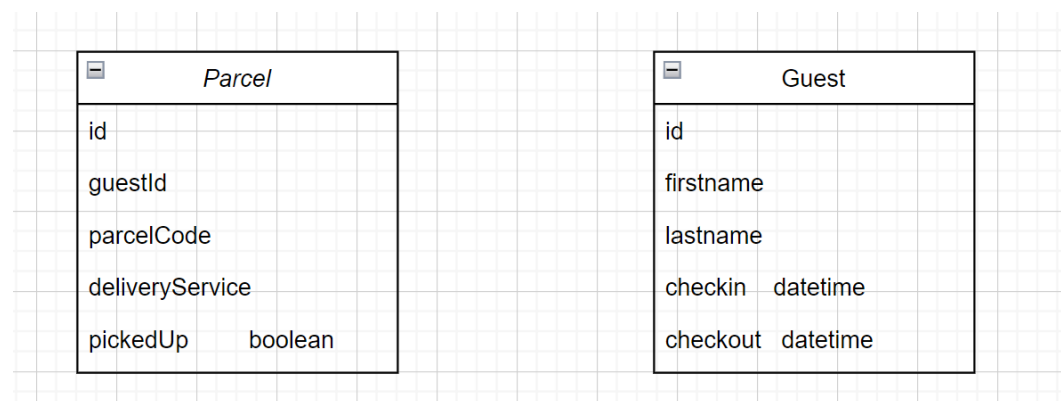


Data model



Swagger API

<http://localhost:8080/swagger-ui.html#/guest-controller/>

<http://localhost:8081/swagger-ui.html#/parcel-controller/>

```
@PostMapping
public String addParcel(@RequestBody ParcelDto parcel) {
    if (!guestClient.checkIfCanAcceptParcel(parcel.getGuestId())) {
        return "Parcel receiver is not a guest or the guest has already checked out. Can not accept the parcel.";
    }
    parcelService.addParcel(parcel);
    return "Accept.";
}
```

```
@PostMapping("/{lastname}/{firstname}")
public String takeParcelForGuest(@Valid @RequestBody ParcelRequest parcelRequest) {
    Guest guest = guestService.getGuestByName(parcelRequest.getLastName(), parcelRequest.getFirstName());
    if (guest == null) {
        return "Parcel receiver is not a guest!";
    }
    if (guest.getCheckOut() == null) {
        ParcelDto parcelDto = new ParcelDto(guest.getId(), parcelRequest.getParcelCode(),
            parcelRequest.getDeliveryService());
        parcelClient.storeParcelForGuest(parcelDto);
        return "Guest is still checked in! Parcel was store";
    }
    return "Guest has already checked out on " + guest.getCheckOut();
}
```

For storing parcel, need to check the guest status first. We will accept the parcel when the guest is checked in and have not checked out.

```

@PutMapping("/{id}/checkout")
public String checkoutGuest(@PathVariable long id) {
    Guest guest = guestService.getGuestById(id);
    if (guest == null) {
        return "This guest can not be found in our system.";
    }
    if (guest.getCheckOut() != null) {
        return "Guest has already checked out on " + guest.getCheckOut();
    }
    guestService.checkoutGuest(id);
    List<ParcelDto> parcelDtos = parcelClient.pickUpParcels(id);
    if (parcelDtos.isEmpty())
        return "No parcel for guest! Guest checkout is successful";

    return "Guest has " + parcelDtos.size() + " parcels stored here!";
}

```

For guests checking out, need to check if they have parcels stored.

Unit tests

