

Strategy:

We go through each features that we want to create from each use cases and then fill up the entities incrementally.

User

- Email: String
- Password: String (contain at least 1 number, 1 capital letter, 1 small letter)
- userRight: Enum
- Profile<?> (1-1)
 - Name: String
 - phoneNum: Long

(0-user, 1-customer, 2- guest, 3- host, 4- admin)

Admin, Tenant, Host

Customer

- Tenant
- Host

Tenant

- Profile<Tenant>
- ~~Venues~~
- Booking : List<Booking> (1-*)

Host

- Listing: List<Listing> (1-*)
- Profile<Host>

Admin

- Email : String
- Password String
- Profile: String

HostDoorSystem

- Host: Host (1-1)
- Listing: Listing (1-1)
- Booking: List<Booking> (1-*)

Profile

- UserProfile
- LocationProfile

Location

- streetName: String
- Lang: Double
- Lat: Double

Listing

- Location: Location (1-1)
- Building type: Enum
- ~~ListingHeader~~
- ListingDetail: ListingDetail (1-1)

ListingDetail

- Header: String
- Description: String
- Price : String
- Photos: String (URL)
- Status: Enum (completed payment, pending, booked, archive, cancelled)(how do we want to make them do the payment - up front payment or deposit system)

Booking

- Location: Location (1-1)
- DateTime: List<DateTime>
- Type: Enum
- Tenant: Tenant (1-1)
- Listing: Listing (1-1)
- Qr-code: String (hashed)

Feedback

- Rating: Int (max is 5)
- Remarks: Enum(Type is it for customer or room)

Transaction

- paymentType: Enum (0 for PayPal)
- Payer: Customer
- Payee: Customer
- Date: DateTime
- amount: BigDecimal
- Currency: Enum (0-Singapore)
- Booking: List<Booking>

PaymentAccount

- Customer
-