Revamp Rezlife

Aubree Lytwyn, Eric Thompson, Jason Hill, Shane Sarnac

<<enumeration>> RoomAttributeState

GOOD
POOR
DAMAGED
MISSING

RoomAttribute

- name: String
- attribute_id: Integer
- status: RoomAttributeState
- description: String
- comment: String

+ getStatus(): RoomAttributeState
+ setStatus()
+ getDescription(): String
+ getComment(): String
+ getName(): String
+ getAttributeId(): Integer

1..*

Room

- residents: ResidentProfile[]
- room_number: Integer
- room_attributes: RoomAttribute[]
- max_occupancy: Integer
- rcf_list:RoomAttribute[]
- residentManager: ResientManager

+ addResident(resident: Resident:Profile): Boolean
+ removeResident(resident: ResidentProfile) : Boolean
+ addRoomAttribute(roomattribute: RoomAttribute): Boolean
+ removeRoomAttribute(roomattribute: RoomAttribute): Boolean
+ isMoveInOk(): Boolean
- generateRCF()

CheckedOut

CheckedIn

Overdue

<<interface>> State

+ charge()
+ getState(): State
+ update(String: checkoutdate, String: duedate)

FrontDesk

- deskitemlist: DeskItem[]
- tempkeys: TempKey[]
- deskid: Integer

1

Item

Building

- roomsList: Room
- name: String
- max_occupancy: Integer
- front_desk: FrontDesk[]
- staff: CU_Job[]

+ getRoomAvailability(): Bool
+ addRoom(Room)
+ getRoomList(): Room
+ getRoom(roomnumber:int): roomAttributes
+ getName(): String

- key_history: Key[]
- desk_item_history: Item[]
- roommate_agreement: RoommateAgreement
- firstname: String
- lastname: String
- email: String
- phonenumber: Integer
- chargehistory: Item

+ addKey(Key)
+ getKeyPresentKey(): Key
+ addRoom()
+ getKeyHistory(): Key[]
+ getPresentRoom(): Room
+ getRoomHistory(): Room[]

0..*

+ modifykey(room: Room, resident: ResidentProfile): Boolean
+ validateIdentity(resident: ResidentProfile): Boolean

ResidentAdvisor

CU_Job

- job_title: String

+ getJobTitle():String
+ CU_Job(int CUId)

CommunityAssistant

BuffOneCard

TempKey

RoommateAgreement

- room: Room
- properties: Boolean

ResidentManager

- roomList: Room[]

+ viewResidentList(): ResidentProfile[]
+ createRCF(room: Room): RoomConditionForm
+ modifyRCF(resident: ResidentProfile): RoomConditionForm
+getResident(resident_id: Integer): ResidentProfile
+ completeResidentCheckIn(resident: ResidentProfile): Boolean
+ completeResidentCheckOut(resident: ResidentProfile): Boolean

FrontDeskManager

- desk:FrontDesk

+ issueTempKey(resident: ResidentProfile, key: Key):Boolean
+ issueOriginalKey(resident: ResidentProfile, key: Key): Boolean
+ removeKeyAccess(key: Key): Boolean
+ checkOutDeskItem(key: Key): Boolean
+ checkInDeskItem(desk_item: Item): Boolean
- chargeTempKey(resident: Resident_Profile)
+ requestDeskItem(resident: Resident_Profile, desk_item: Item[])
- isIdentityValid(resident: ResidentProfile, key: Key):Boolean

RoomConditionForm

- complete: Boolean
- room_attribute_list:I list
- Cu_id

1..*

Demo :

USR-012
As a CA, I must be able to check out an item to a Resident.

Design pattern: state

Why…?