Space - 85E - 010  Schemas  Former methods  Assignment 03  [introducing sets]  [introd	V	Hazi's Ej	iaz		11th dec 2021
Former methods  Assignment 03  [introducing sets]  [introducing se				Schemas	
Assignment 03  [introducing sets]  [introducin					
TotalRooms     Introducing sets      Room_STATUS ::= VACANT   OCCUPIED      RESPONSE 11 = Success   atready_added   atready_occupies   atready_Vacant   not Insystem      (declosing free types)	-				
RESPONSE 11 = SUCCES   atready added   atready occupied   atready Vacant   not En System  (declaring free types)  Lecturing free types)  Commission & commission OK V atready added  Cocupy & occupyok V not In System V arracky occupied  Vacate & vacate ok V not In System V atready vacant  (state schema)  System  Rooms: P TextRooms	-	Hasignmen	k 03		
RESPONSE 11 = SUCCES   atready added   atready Occupied   atready Vacant   not En System  (declaring free types)  3: (use case scenarios)  .: commission & commission OK V atready added  .: occupy & occupyok V not In System V arraxly occupied  .: Vacate & vacate OK V not In System V atready vacant  (state schema)  System  Rooms: P Toul Rooms	-	<u>!</u> [	Totalk	Rooms 7 (introducing sets)	
RESPONSE 11 = Success atready added atready occupied atready Vacant (not Ensystem  (declaring free types)  (use case scenarios)  Commission & commission OK V atready added  coccupy & occupyok V not-In-system V arready occupied  Vacate & vacateok V not-In-system V already vacant  (state schemo)  System  Rooms: IP Total Rooms				t whalk	
RESPONSE 18 = Success atready_added atready_occupied atready. Vacant (not Ensystem  (declosing free types)  3: (use case scenarios)  .: commission & commission OK V atready_added  .: occupy & occupyok V not_In_system V arready_occupied  .: Vacate & vacate OK V not_In_system V atready_vacant  (state schemo)  System  Rooms: IP TotalRooms	3	2; Roon	n_ STATU	4 := VACANT   OCCUPIED S	
(declosing free types)  3: (use case scenarios)  .: Commission ≥ commission OK V already_added  .: occupy ⊆ occupyok V not_Im_System V asready_occupied  .: Vacate ⊆ vacateok V not_In_System V already_vacaut  (state schema)  System  Rooms: IP TockRooms		RES	PONSE	11 = Success aready added already occupied already vacant not	.En.System
3: (use case Scenarios)  Commission ≥ commission OK V atready_added  Cocupy ⊆ occupyok V not_In_System V arready_occupied  Vacate ⊆ vacate OK V not_In_System V already_vacaut  (state schema)  System  Rooms: IP TotalRooms	•				
Commission ≥ Commission OK V already_added  Coccupy ⊆ occupyok V not_Im_System V already_occupied  Vacate ⊆ vacate Ok V not_In_System V already_vacant  (state schema)  System  Rooms: IP Total Rooms	-			( Carrier )	
Commission ≥ Commission OK V already_added  Cocupy ≥ occupyok V not_Im_System V already_occupied  Vacate ≥ vacate Ok V not_In_System V already_vacant  (state schema)  System  Rooms: IP Total Rooms		3:	( use a	ale Scenarios)	
∴ occupy   occupy   occupy   occupy   vacate					
Vacate   Va	*				
System  Rooms: IP Total Rooms	Distant		-		8
System  Rooms: IP Total Rooms	*		السرز -	· Vacage on	
System  Rooms: IP Total Rooms	4	<u>ч.</u>	(state	SChema)	
Rooms: IP Toral Rooms		-			
	-				
100m specification . Her	-				
	_		100m - p.	eci Ficación : Haves	
S; (Init state)			17.14	and a	
	4				
Init_					
System  Rooms - P (roomspecification is automatically NULL since Rooms = P)				,	
Rooms = \$\Phi \text{(room-specification is automatically NULL since Rooms = \$P\$)}	-		Koons	- William absention in the many and a second	
by (use case : commission)	_	he	. (1	use case : (omnission)	
CommissionOK	-	,			
Δ system	-16	,			
newRoom? : TotalRooms; Tes! : RES PONSE	-				
now Room? & Total Rooms					
TOOM Specification'= Toom Specification U ( nawRoom? -> VACANT)			- Spi	-: Lination = Toom Specification U (NawRoom? -> VACANT)	
Yes! = succes	-		-		
	-		1-1-		
	5	-			
	-				

80	11/1	

	_=	
<u>*</u>	(usecase: occupy)	ik s grad dije 🗨
	D ccupy ok	9
	△ System	•
		A roll 1)
	Toom? E Total Rooms	
	Yourspecification (Your?) = VACANT	
	room specification = room specification & (room? > occupied)	n 10.1931
	Tes! = success	
	Usecase: vacate)	
	V v	
	Vacale OK	1 1 2
	△ System	The second second
	input Room? : Total Rooms	
	TES! : RESPONSE	•
	input Room 3 & Total Rooms	CHANGE OF THE PARTY OF THE PART
	Your specification (input Rown?) = OCCUPIED	
	Toom Specification = room Specification @ (input Room? -> VACANT)	•
	Tes! = Success	lat.
		•
a;	(query to count occupied rooms)	
	Decupied Court_	
	8 System	
	Occupied Rooms : P Total Rooms	
	Occupied Count: N	
	occupied ( out = # ( Room Specification ( Occupied Rooms) = Occupied)	
	(Steaphadicond) = Occupied)	
		-
		•

Egron Scenarios
Already Added
□ System
Youn? : Total Rooms
Yea! : RESPONSE
Youn? E Total Room
Tes! = Already Added
not In System
S System
Yourn ?: Total Rooms
Yes! ; lesponse
Youn? 4 Total Rooms
res! = not In System
Tes; - not 14 system
3; Ali cody Vacort
E System
Toom? : Total Rooms
res! : RESPONSE
Toom Specification (room?) = VACANT
res! = Already Vacout
Already Occupied
□ System
room?: Total Rooms
res! : RESPONSE
roum Specification (room?) = OccupiFD
res! = already Occupied