

# Software Engineering

## Deliverable 04

---

*Equivalence Class Partitioning &  
Boundary Value Analysis on One Use Case*

### Restaurant Management System

Prepared By:  
Muhammad Daud Mazhar 18L-0919  
Zulfiqar Chaudhry 18L-1037  
Muhammad Ahmad 18L-0965

NUCES FAST, Lahore  
BCS 6D  
25 June, 2021

## Use Case:

The use case implemented here consists of an incoming customer looking at the available table interface. The customer is then prompted to select an available table for booking. For the implementation, we have limited the no of tables from 1 to 6. Only vacant tables can be selected, otherwise an error will be generated.

Floor Layout	
Table No: 1 Status: ``Vacant`` Chairs: 4	
Table No: 2 Status: ``Occupied`` Chairs: 2	
Table No: 3 Status: ``Dirty`` Chairs: 2	
Table No: 4 Status: ``Vacant`` Chairs: 6	
Table No: 5 Status: ``Vacant`` Chairs: 6	
Table No: 6 Status: ``Dirty`` Chairs: 2	

## Equivalence Classes:

Equivalence class 1: Table no below 1

Equivalence class 2: Table no from 1 to 6

Equivalence class 3: Table no above 6

## Boundary Values Test Cases:

Test case 1: Table No 0: Member of equivalence class 1 and adjacent to boundary value.

Test case 2: Table No 1: Boundary value.

Test case 3: Table No 2: Adjacent to boundary value.

Test case 4: Table No 4: Member of equivalence class 2.

Test case 5: Table No 5: Adjacent to boundary value.

Test case 6: Table No 6: Boundary value.

Test case 7: Table No 7: Member of equivalence class 3 and adjacent to boundary value.

## Results for Test Cases:

Test Case 1: Table No 0  
(Invalid Table No)

```
Please enter a Table No to book: 0
Error! Invalid selection. Enter positive integers from 1-6:
```

Test Case 2: Table No 1  
(Vacant Table Booked)

```
Please enter a Table No to book: 1
Success! Your table no: 1 has been booked!
```

Test Case 3: Table No 2  
(Valid Selection but Table  
not Available)

```
Please enter a Table No to book: 3
Sorry, this table is not vacant yet. Please try another table:
```

Test Case 4: Table No 4  
(Vacant Table Booked)

```
Please enter a Table No to book: 4
Success! Your table no: 4 has been booked!
```

Test Case 5: Table No 5  
(Vacant Table Booked)

```
Please enter a Table No to book: 5
Success! Your table no: 5 has been booked!
```

Test Case 6: Table No 6  
(Valid Selection but Table  
Not Available)

```
Please enter a Table No to book: 6
Sorry, this table is not vacant yet. Please try another table:
```

Test Case 7: Table No 7  
(Invalid Table No)

```
Please enter a Table No to book: 7
Error! Invalid selection. Enter positive integers from 1-6:
```

**Note:** The source code for this use case implementation is attached in the .cpp file.