**\*\*\*\* CRC CARDS\*\*\*\***

|  |  |
| --- | --- |
| **Class Customer** | |
| **Responsibility** | **Collaborator** |
| Sign up for a table |  |
| Receive notification for available | Restaurant |
| Confirm Reservation | Restaurant |
| Cancel Reservation | Restaurant |
| Get total visitor along with the main customer |  |

|  |  |
| --- | --- |
| **Class SignUp** | |
| **Responsibility** | **Collaborator** |
| Gets reservation details from customers | Customer |

|  |  |
| --- | --- |
| **Class ReservationHandler** | |
| **Responsibility** | **Collaborator** |
| Accept table booking request from Customer | WaitListHandler |
| Notify Customer for availability of table |  |
| Allocate table to the eligible customer | Customer |
| Get available table status | TableHandler |

|  |  |
| --- | --- |
| **Class WaitListHandler** | |
| **Responsibility** | **Collaborator** |
| Add customer to waiting queue | ReservationHandler |
| Remove customer to waiting queue | ReservationHandler |

|  |  |
| --- | --- |
| **Class TableHandler** | |
| **Responsibility** | **Collaborator** |
| Get unoccupied/available table | Table |
| Get unoccupied/available table chair count | Restaurant |

**Design Patterns:**

**Chain of responsibility Pattern:**

WishListHandler will handle all the TableList and CustomerList. It will match the customers and TableList according to the table and customer party size.

**Singleton Pattern**

Since it is possible to have only one instance of restaurant class and all customers should be added to this single instance of Restaurant class.