Group 1 Team 2 – Ben Xerri, Jagdeep Singh, Kwasi Young

|  |  |  |
| --- | --- | --- |
| **Entity Class:** | **Attributes** | **Description** |
| Order | Order\_ID  Price  Time\_Placed  Table Number | After interacting with the table, the server creates an order in RAS. The order is for the entire table consisting of a unique identifier, number of items, price, time placed. |
| Order Queue | Order Queue []  Number of Items(Total length of Queue) | A list containing the order objects that have been created by the place order function. |

|  |  |  |
| --- | --- | --- |
| **Boundary Class:** | **View** | **Description** |
| Server Interface  -Team | |  | | --- | | placeOrder | | updateOrder | | viewOrder | | payOrder | | Upon logging in, the server will have a unique view allowing them to choose from their 4 functions. placeOrder, updateOrder, viewOrder, and payOrder. |

|  |  |  |
| --- | --- | --- |
| **Control Classes:** | **Actor:** | **Description** |
| placeOrder  -Ben Xerri | Server | After the creating the order with the table, placeOrder submits the order to the kitchen queue. After an order is placed, it can be viewed by the viewOrder function. |
| viewOrder  -Ben Xerri | Server, Manager, Chef | After the order is placed, it is collected in a list and stored on a first in first out basis. When an order is in the queue, it can be viewed by the manager, server, and chef. When an order Is prepared, it is removed from the front of the queue. |
| payOrder  -Jagdeep Singh | Server | Once the Guest is ready to check out, the server is able is see all orders using viewOrder and clear the total for a particular order/table. The order status must be cooked in order for payOrder to occur. |
| updateOrder   * Kwasi Young | Server | Represents the function that allows the items of the order to be edited. |

Order Management Subsystem: - Ben Xerri

Along with the server; the manager and chef employee’s will need to use the viewOrder. The chef needs to viewOrder before cooking the meal. The manager needs to viewOrder to assist other employees.

In general, this order system may be useful in any system that needs to implement a queue based service, such as an automated system that calls customers to be assisted based on the order they arrive.

**Order - Dynamic Model**

Ben Xerri

viewOrder_v2%20(1).png

placeOrder_SE%20(2).png

Jagdeep Singh – payOrder



Kwasi Young – updateOrder

Services Required: Requires Orders data that will be used to display. [This is precondition, not required service. You may use ‘None’.]

Services Provided: Provides data for the orderList that displays in the Kitchen. [This should be “Allow updates for a created order. The constraint: the items in the order cannot be updated if the order status is ‘cooked’. ]

collaboration diagram.png

Kwasi Young

**Subsystem Interfaces**

Service List