Bowl Backend System:

1) Web API:
    a) **send_status()** - sends most recent system status (temp, ingredients, order status etc) to the main server. *Repeat every 5 mins.*
    b) **get_order()**: response to send_status() if there is an order, returns an accept or reject based on whether lockers are expected to be full at a certain time.
2) Back-End Files:
    a) **set_machine_config.txt:** Ingredients to pod relation
    b) **current_status.json**: Current temperature, arduino status, order_status
    c) **order_queue.json**: total order queue arranged by time
    d) **temp_order.json**: temporarily store an order till it is added to the queue
3) Back-End Algorithms:
    a) **path_time_estimate() -** Calculates the path & time for an order
    b) **order_queue()** - queues the orders based on the collection time/order method
    c) **locker_queue() -** assigns a locker to an order if more than one order at a time
    d) **update_config()** - subtract material from ingredients if order complete
    e) **start_order()** - run the order at the current timestamp based on order_queue.json
    f) **start.sh** - one click start for the backend
4) Back-End Interface:
    a) **home.html** : backend home
    b) **generic.html** : backend set configuration (pod-ingredients matching)
    c) **elements.html** : backend web elements for future developers

5) Arduino API:
    a) **get_status()-** requests the arduino for the current status. *Repeat every 30 secs.*
    b) **locker_status()** - *checks the locker status (is picked?)*
    c) **open_locker()** - *open a certain locker*
    d) **make_salad()** - *make an order*
    e) **order_complete()** - *wait for an order to complete & put in locker*

Path_time_estimate: finds the shortest path - goes row by row, each row starts from the closest point where the previous row ends. (Repeats 3 times for “3D printing”)

Order_queue: Arranges orders in increasing order of *start_time=pickup_time - preperation_time* under the constraints:

- Orders made at the machine get first priority

- Two orders cannot be made at the same (delivery to locker time adjusted)
- Orders with clashing times are ordered back to back, with shorter preparation time being prepared last
- If the lockers are expected to be full at a certain time, we return a “FULL, ORDER REJECTED” status to the Web API