Restaurant Functional requirements:

Overview

The simulator will require the ability to simulate the passing of time(minutes/hours/days) to allow:

- Tabs to expire.
- Waiters return to tables after a timer.

- Customers' moods change based on time taken for order to return or waiter to return.
- Order to be processed within a limit to allow changes in customer state based on time taken

The Floor:

The floor represents one of 2 primary components we will build our simulator from

The Floor must:

- Have and maintain a max capacity✔
- Allow the construction and destruction of tables
- Maintain a list of “customer bills”.
- Allow users to navigate to available
- Allow the maitre D to allocate customers to tables.✔
- Let users merge and split tables if necessary --?
- Allow the floor to be split into subsections allocated to waiters
- Allow users to view the full floor
- Make the decision to Reserve a table if booking is made.✔
- (make the decision to allocate an opening and closing time)

The engine:

- Will be in charge of communicating updates to all timers by notifying classes of changes

A maitre D:

- Make decisions for merging and splitting tables based on party numbers
- Make the decision on allocating Reserved tables.
- Make the decision to split tables and link with the corresponding waiter
- Allocate tables to customers minimizing waste (for party of 4 if table 1(10 seats) and table 2(6 seats) allocate table 2.
- Maintain a queue of parties waiting to be seated. Refer to the floor to validate reservations
- Allocate a waiter to a table
- Make the decision to stop allocating tables based on max capacity
- Maintain a list of waiters

A Floor manager: x

- Maintain a list of currently active Waiters
- Maintain a list of currently active Tables
- Occasionally traverse the floor

The customer:

- Make decisions of tip based on their current mood.

- Create orders with custom decorators.
- Make decisions on creating, or removing “bills” (if paid).
- Notify the table when they are ready to make an order.
- Have the ability to set a custom tip.
- Make the decision to pay a bill or split the bill into sub bills (allow the customer to either select the % split or custom split amounts).
- Make a complaint if mood becomes “angry”.
- (create mood conditions based on time taken for order/order accuracy/time taken for waiter to return/random mood change)
- Notify the table when they are ready to leave.
- (Make the decision to leave if they have not been allocated a table within a time)
- randomly set readyToOrder to true, with bias when waiter comes to check order

The table:

- allow only tables of 1, tables of 2 and tables of 4.
- Be allocated a waiter.
- Make the decision to notify the waiter that all customers are ready to order/set timer for waiter to return to check if customers are ready to order
- Be allocated a floor section/sub-section.
- Merge into n tables if notified
- Split into n tables if notified

- Maintain a list of outstanding orders and completed orders.
- Maintain a list of currently connected tables in the 4 cardinal directions.
- Change state to “empty” when customers vote to leave.

The Waiter:

- Indicate the section of floor or table they are assigned. (by the Maitre D)
- Maintain a list of tables with outstanding orders/haven’t ordered yet.
- (Maintain a timer for each table to simulate returning “after a while” if indicated by the table.)
- Maintain tables bills (tab bill, full bill, split bill)
- Make a decision on the customer's bill if the bill's expiration date has been reached or has passed.
- (have a concrete time to decide whether to traverse their entire section or individual timers for individual tables)
- Traverse the floor

The kitchen

The kitchen represents the second of the 2 primary components

The kitchen:

- Maintain a list of currently active chefs.

- Maintain a list of chef responsibilities, each to be implemented or passed as part of the chain.
- Allow the interaction of a waiter and head chef over an order or a single dish
- Allow chefs, managers, and waiters to do rounds.

The head chef:

- Be the beginning and end of the dish chain.
- Make the decision as to when each of the other stations will receive the dish.
- Notify the waiter when a dish is complete.
- Occasionally traverse the floor

A chef:

- Determines if the dish they have been passed requires decoration.
- Pass the dish to the next chef as indicated by the dish.
- Pass the dish back to the head chef when/if complete.
- Have the ability to adjust preparation based on preferences.

A dish:

- Have cooking decorators.
- Maintain a reference to the customer that ordered it(create Dish “state” after cooking, undercooked, perfect, overcooked may affect customer mood

differently, implemented randomly to allow variation with bias)
    - allow attribute “time taken” to allow variation of customer mood if too much time is taken

A menu:

- Allow customers to decorate a dish using sides, drinks.
- Allow customers to remove from dishes based on allergens/preferences.
- (show a custom menu if customer indicates allergy) 😵
- Manage the creation/destruction of dishes
- Assign unique preparation times based on item type (starters have shorter times etc.)

A bill:

- Have the ability to adjust cost based on the presence of a tab.
- Allow more than one way to pay (card, cash) (may require chain to determine change to return)
- Make the decision to split into multiple bills when prompted.
- be dynamically allocated based on the orders, decorations and omitted items.
- account for customer tips given based on satisfaction.