**Restaurant Manager**

1. Introduction
   1. The system "Restaurant Manager" is an application designed to assist managing a restaurant with hundreds of customers every day, and a large menu.
   2. The acting manager, Yossi Salat, wants a system that will help him manage the restaurant by providing the following:
      1. Easy to use digital menu for quick and accurate order time, including different types of the same dish.
      2. Streamline communication between waiters and food preparation.
      3. Enable customers to split the bill easily providing different payment methods.
      4. Provide statistical information about the restaurant's performance for future advancements.
2. Functional requirements
   1. Initialize/change restaurant
      1. Add/remove table
      2. Add/remove dish
         1. Add/remove dish extras
      3. Add/remove stations
      4. Alter number of waiters
   2. Save/load restaurant parameters
      1. Save/load the current status of:
         1. Tables
         2. Menu
         3. Stations
         4. Number of waiters
   3. Seat new customers
      1. When a group of customers arrive, a waiter accepts the group and gives them a table. The waiter records the number of guests and time of arrival for statistical analysis.
   4. Take order
      1. Take the order from each customer at the table creates a new unique order in system tied to the table.
      2. Issue tasks to relevant stations, and notify the stations.
      3. Record the orders.
   5. Change order
      1. If the order hasn't been delivered to the table, allow the customer to change the order.
      2. Issue updates to relevant stations and notify the stations.
   6. Cancel order
      1. If the order hasn't been delivered to the table, allow the customer to cancel the order.
      2. Notify the relevant stations about the cancellation.
      3. Record that the order has been canceled.
   7. Prepare dish
      1. When a station receives a new order or order change, begin preparing the dish, and mark the dish as being prepared.
      2. When a station receives an order cancellation, cancel the order.
      3. When a dish is prepared, mark it as ready. If this is the last dish to be prepared mark the order as ready, and notify the waiters.
   8. Supply order
      1. When an order is marked as ready, take it to the table. The order is marked as delivered.
      2. Record the time of delivery.
   9. Finish meal
      1. The system prints a receipt for the table for the relevant order, and deliver to the table.
      2. Record the time of delivery of the receipt.
   10. Take payment
       1. Each customer may pay for himself or for the whole table.
       2. The customer can pay in three ways:
          1. By cash
          2. Credit card
             1. Verify the card with credit card company
          3. Application .
       3. Print receipt if there was no use of the application .
       4. Record the payment methods and amounts.
3. Non-functional requirements
   1. Performance
      1. The system should be able to scale the amount of waiters/dishes/tables without effecting customer satisfaction
   2. Portability
      1. The system should be able to apply to any different restaurant requirements.
4. System evolution
   1. Require minimum charge for certain payments.
   2. Allow additional payment methods.
   3. Allow customer built dishes.
   4. Allow customer to share dishes to social applications.
   5. Allows for cancelled dish that has been prepared to be reassigned to a different order when possible.