The Restaurant Manager

1. Introduction

    1.1. The Restaurant Manager (henceforth referred as “the system”) is a software application intended to assist in running and managing a restaurant with multiple service types and employees.
    1.2. Yossi Salat is a restaurant owner who wishes to have a system that will aid the restaurant’s staff in the following manners:
        1.2.1. Improve communication between waiters and kitchen,bar & pastry bar.
        1.2.2. Reduce customers waiting time caused by ordering dishes that are currently unavailable.
        1.2.3. Reduce customers waiting time caused by ready orders that remain undelivered.
        1.2.4. Reduce complexity of billing one or more people who pay for the table.
        1.2.5. Ease selection of particular variations of certain dishes.
        1.2.6. Increase flexibility in modifying existing orders.
        1.2.7. Enhance long-term planning with statistical analysis of restaurant’s routine.

2. Overall Description

    2.1. The system shall provide different graphical user interfaces, customized per roll, which will enable an employee to easily sync the rest of the staff about his tasks’ statuses.
    2.2. Data about orders (including unsatisfied orders) and current amount of diners will be regularly collected and analysed by the system.
    2.3. Changes in menu items are made by managers.
    2.4. Managers will have multi-functional access to the system, and they will be able to update any employee controlled field at any time.

3. Functional Requirements

    3.1. Accepting a new diner
        3.1.1. When a diner group (or single diner) arrives at the restaurant, a waiter accepts the group, records the number of guests is the system (and the time they arrived).
    3.2. Ordering meals
        3.2.1. Upon a table ready to order, the waiter takes the order, creates a new order in the system and updates its details.
        3.2.2. In case a part of the order is unavailable at the moment of inserting it to the system - the waiter is notified and requested to update the diner. The rest of the order will be processed, and there will be no tracking of the declined order.

3.2.3. Tasks will be issued to relevant employees, who will be notified.

3.3. Prepare dish

3.3.1. A bartender / cook / confectioner receives a task from the system.

3.3.2. Upon completion of a task - if an employee decides a dish can't be ordered in the future (for example - an ingredient is missing) - he will mark the menu item as unavailable. This action will take place prior to 3.3.3.

3.3.3. Upon completion of the task - its status will be updated by the relevant employee and the system will determine if an order is ready to be served (i.e. - all items in order are marked as ready).

3.3.4. When an order is ready - a waiter will be notified.

3.4. Deliver order

3.4.1. When receiving a notification about a ready order, the waiter will deliver the order to the right table.

3.4.2. After delivery - the status of the order will be updated accordingly.

3.5. Modify order

3.5.1. If a client decides to change his mind before the order is marked as served, the waiter will modify the order accordingly once notified, as following:

3.5.1.1. A modified item's new status will be saved, the old status will not be kept. The relevant employee will be notified.

3.5.1.2. If a new ites is ordered, all relevant information will be stored as in 3.2.

3.5.1.3. A cancelled Item will remain in the system for stats, but will be marked as cancelled. The relevant employee will be notified, and will not be able to change it's status. This action is not reversible.

3.6. Perform payment

3.6.1. When a waiter is asked for the check - the system will produce one, describing the entire order details.

3.6.2. Receiving payment can be done in two methods:

3.6.2.1. Cash - cashier status is updated.

3.6.2.2. Credit card - external unit is activated to contact credit card company, and sends back approval status.

3.6.3. Payment can be accepted separately from each diner.

3.6.3.1. A single Diner can split the amount between the two payment methods.

3.6.4. A receipt will be produced per paying transaction, and the order's account will be updated accordingly.

3.6.5. After payment the order will be marked as paid and no further actions will be performed on it.

3.7. Get statistics report

3.7.1. A manager can request a statistics report for the restaurant.

3.7.1.1. After creation - the report will be displayed on the monitor.

3.7.1.2. There will be shortcuts for the following report types:

3.7.1.2.1. Busy hours report.

3.7.1.2.2. Items popularity report.

3.7.2. Statistics will be kept available until system DB is reset.

3.8. Update / initialize system

3.8.1. A manager can initialize system or change it's parameters.

3.8.1.1. Modifiable fields include:

3.8.1.1.1. Available menu items and prices.

3.8.1.1.2. Available menu items' variations and prices.

3.8.1.1.3. Amount of guests.

3.8.2. A manager will also be able to reset statistics database.

3.8.2.1. Upon DB reset - a backup of the existing DB will be saved as an external file. The DB itself will be cleared.

3.8.2.2. An old backup file can be loaded by manager.

4. Non-Functional Requirements

4.1. Performance

4.1.1. The system's response time will not be affected by the existing amount of open orders.

4.2. Portability

4.2.1. The system will be implemented in a way that will allow rebasing on a different platform without any necessary action to be taken by end-users.

4.3. Connectivity

4.3.1. All system parts will be connected via wired/wireless local network.

4.3.2. There will be a central database, which will be updated per action.

4.4. Security

4.4.1. Interface with external service providers (credit card companies/ banks) will be encrypted and monitored by 3rd party software.

4.5. Survivability

4.5.1. Central DB will be backed up every configurable amount of hours.

4.5.2. Backup copies of the central DB will be kept as external files, for survivability alone - no actions will be performed over them.

4.6. Concurrency

4.6.1. Parallel access to main DB will be allowed and supported.

4.7. Ease of Access

4.7.1. An Item order placement will not take any more than 5 screen taps.

5. Future Requirements

5.1. Support for mini-portable devices for waiters to carry with them, that will gradually replace the static post used presently.

5.2. Add interactive interface for guests at the table. The interface can include images of dishes, recommendations and a "call the waiter" button.

5.3. Add automatic indication of unavailable menu items based upon missing ingredients, so the update won't be done manually like now.