Request For Proposal (RFP)

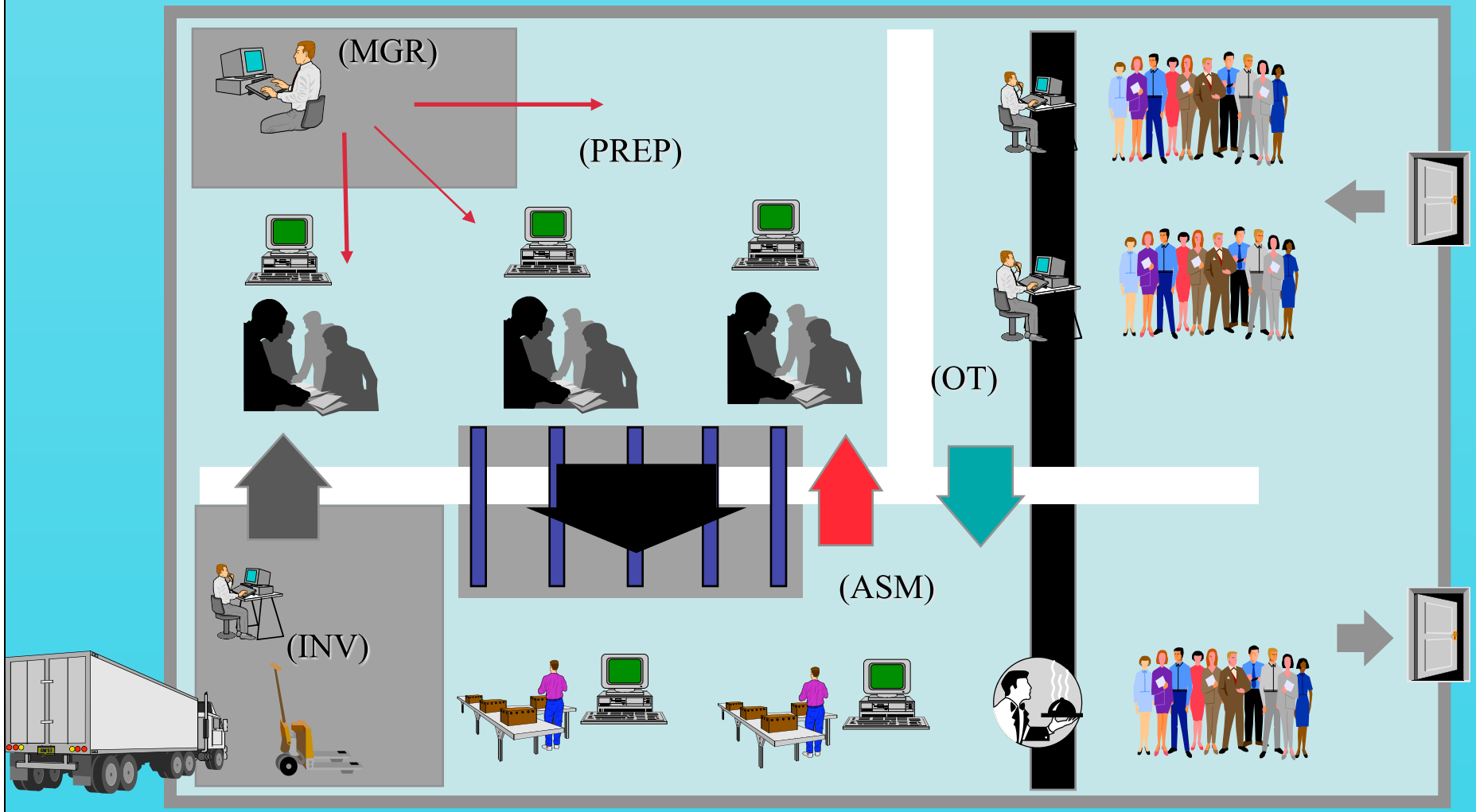
MacFood restaurant system

MacFood is a new restaurant chain which offers fast food to the customers. It uses an in-store computer system to assist order-taking and payment, food preparation, delivery, and inventory.
Orders and payments are taken by staff using “touch-screen" displays.
Kitchen and delivery staff view orders on displays, and register the status of orders by pressing buttons of the keypads.
Inventory of the food and supplies is tracked by the computer system.
The restaurant manager is able to configure the system to set menu items , ingredients, prices, inventory levels, and store setup.
The following section briefly introduces the various units of the MacFood System.

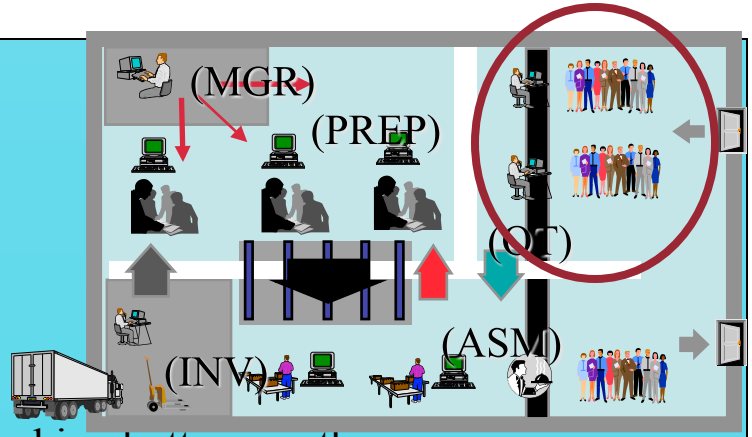
The following slides discuss the produced SRS after requirement analysis phase

Fast Food Restaurant System

Physical view

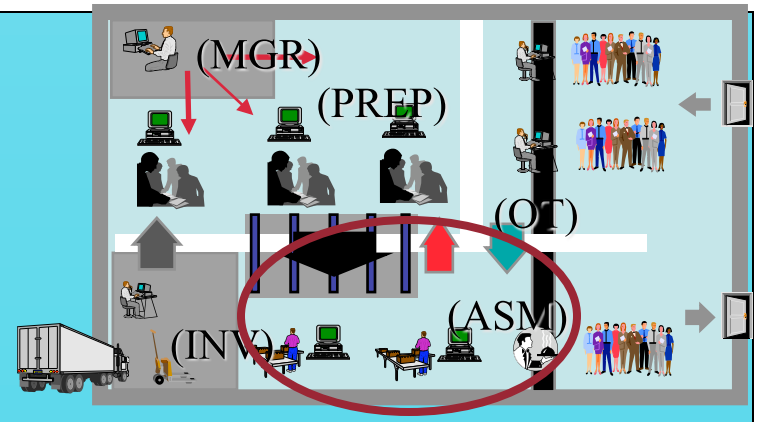


Order-Taking Unit



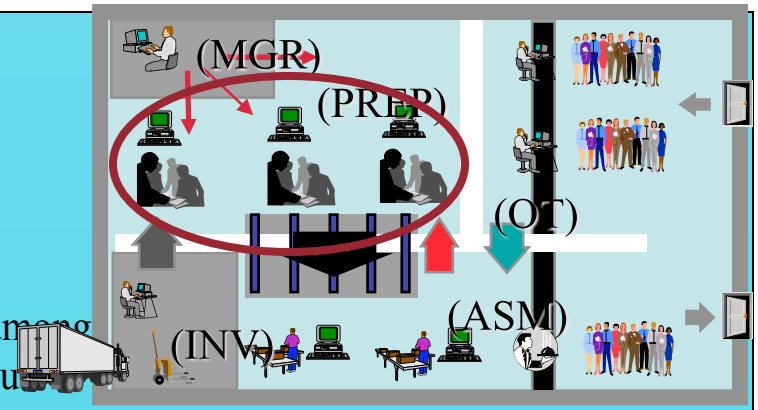
- This unit sets up customer orders and handles payment.
- Menu items are selected from the restaurant-menu by touching buttons on the touch-screen.
- Selection of an item causes it to be added to the current order (which is displayed in a scrollable window on the screen), and the subtotals / tax of the order are displayed.
- An order can be paid anytime between its set-up and delivery to the customer.
- The system keeps the cash balance of each order-taking station and has facilities for supporting "cash float" (i.e., a specified amount of cash in the order-taking station at the beginning) and "skim" (i.e., a threshold amount of cash, which once exceeded, must be transferred to the cash balance) of each station.
- Each order is handled by only one order-taker; however, the orders could be stored in a list and each order-taker in the system can access this list to service the stored orders.

Assembly Unit



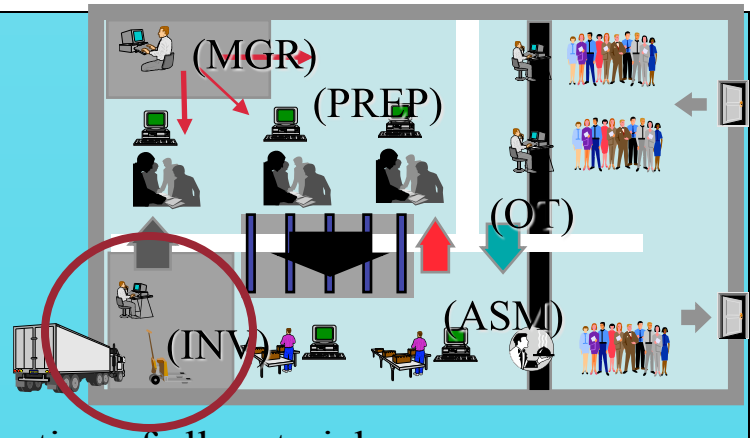
- When an order is set up, the kitchen should be informed to prepare the order-items.
- When the computer system determines that all items of an order are available in the chutes, the order can be assembled.
- Each available assembly-station picks the order and displays it on its screen.
- The assembly-stations use screen and keypad for interaction with the staff.
- The staff assemble the orders, and using keypads inform the system. If the order is paid, the system allows the delivery of the order to the customer, otherwise, the delivery will be postponed to the time that the order is paid.
- If the system indicates that an order can be filled, but the chutes do not contain a sufficient quantity of some order's item, the staff report the shortage to the system to be prepared.

Food Preparation Unit



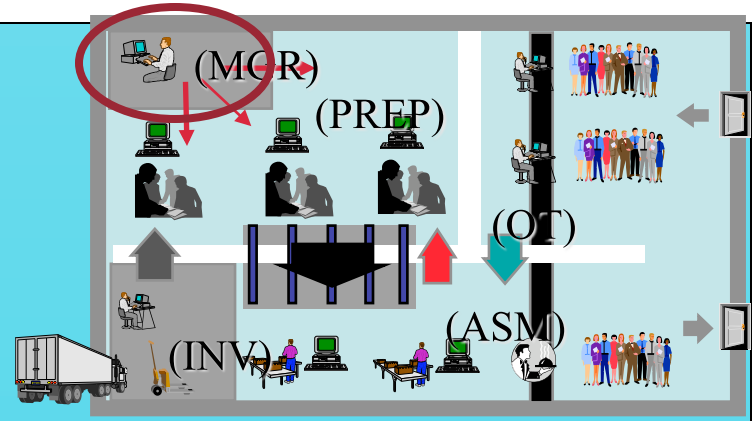
- In order to prepare an order, the system distributes order-items among stations, equipped to prepare certain items of the restaurant-menu.
- In general, more than one station is capable of making a particular item. Each station has a screen and a keypad. Similar items of different orders are grouped together.
- Considering the number of items assigned to each station and its current load of work, the system decides whether to send the items to that station or not.
- The screen of the preparation-station displays a list of items and their quantities.
- Kitchen staff prepare the required quantity of an item, put them in the "chute", and using the keypad inform the system.
- There is one chute for each menu item.
- Menu items are prepared in response to real and anticipatory demands. Anticipatory demands are set up by the manager to shorten the average time of waiting for food.

Inventory Unit



- The inventory unit in the system keeps track of the consumption of all materials used for preparation and packaging of the order-items.
- We refer to these materials as "raw-materials". This unit has a very close interaction with the preparation unit.
- The system keeps stock, and the inventory of raw materials is updated dynamically.
- The arrival of new materials into storage is entered into the system by the staff, and the consumption of the materials is dictated by the recipes of food-items.
- To preserve stock integrity, the system assumes a minimum threshold for usage of each menu-item in the system. If the number of a certain menu-item drops below this threshold, it is considered unavailable and the inventory unit alerts the order-taking unit to inhibit taking that item.

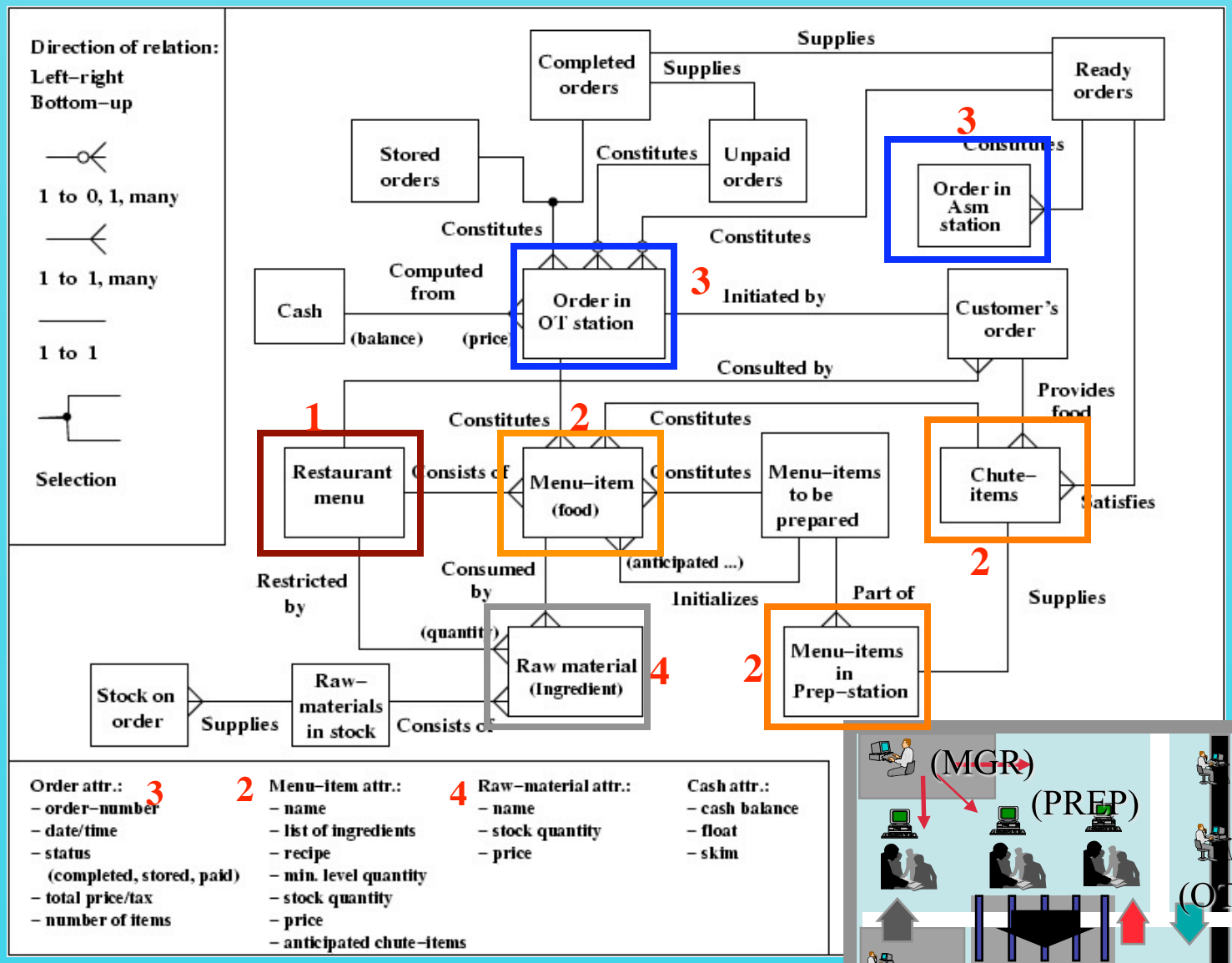
Management Unit



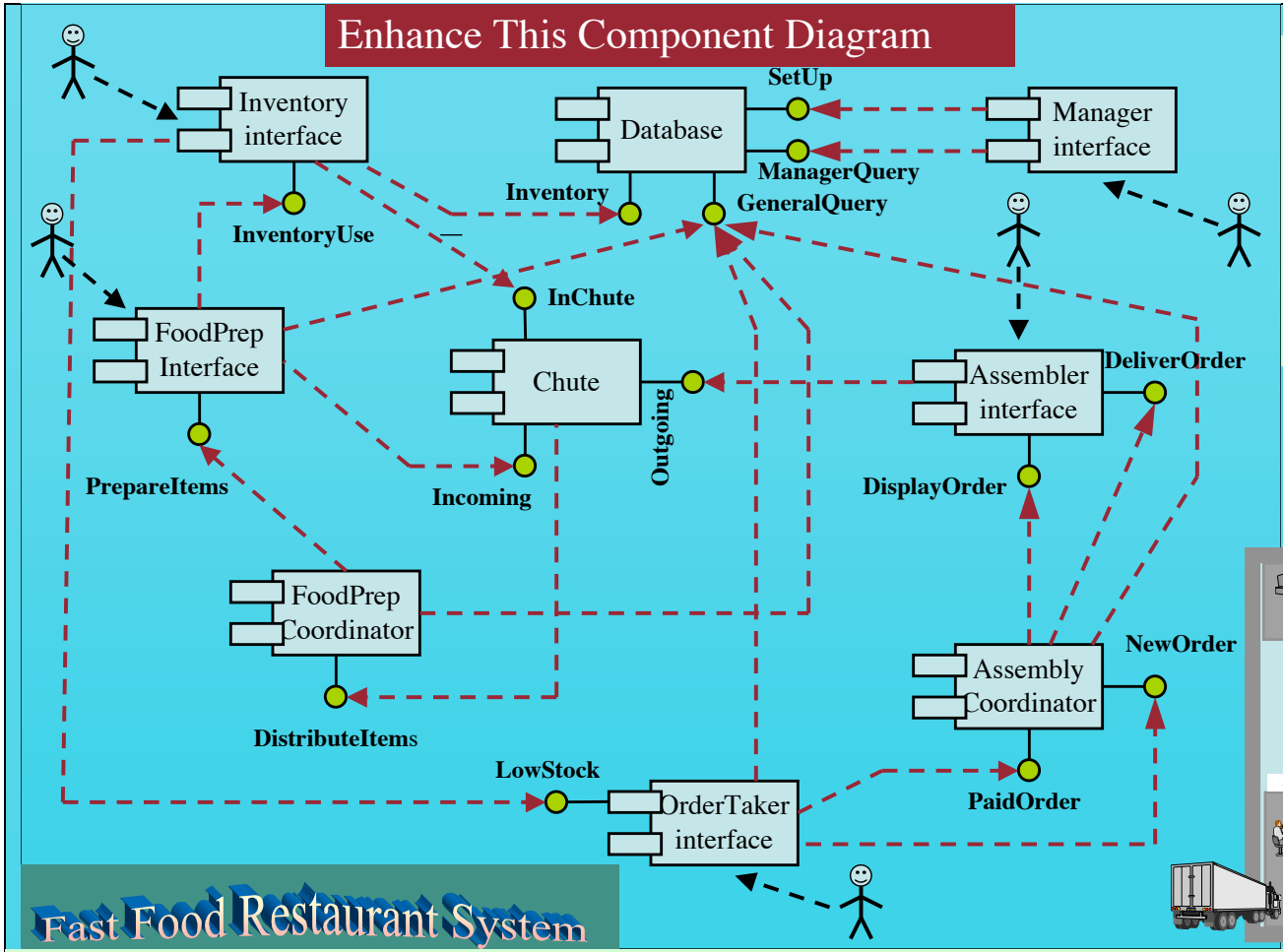
The management-unit of the restaurant system is responsible for setting up:

- Active stations in order-taking, preparation, and assembly units.
- System tables such as restaurant-menu, recipes, anticipated demands, minimum number of menu-items, and raw-materials in stock.
- List of menu-items to be prepared by each preparation station.
- Cash skim and float.
- Different applicable taxes.
- System time and date.

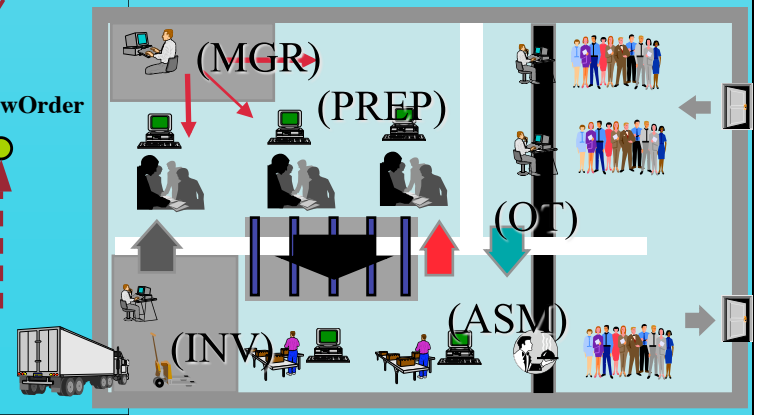
E-R diagram of the Restaurant System



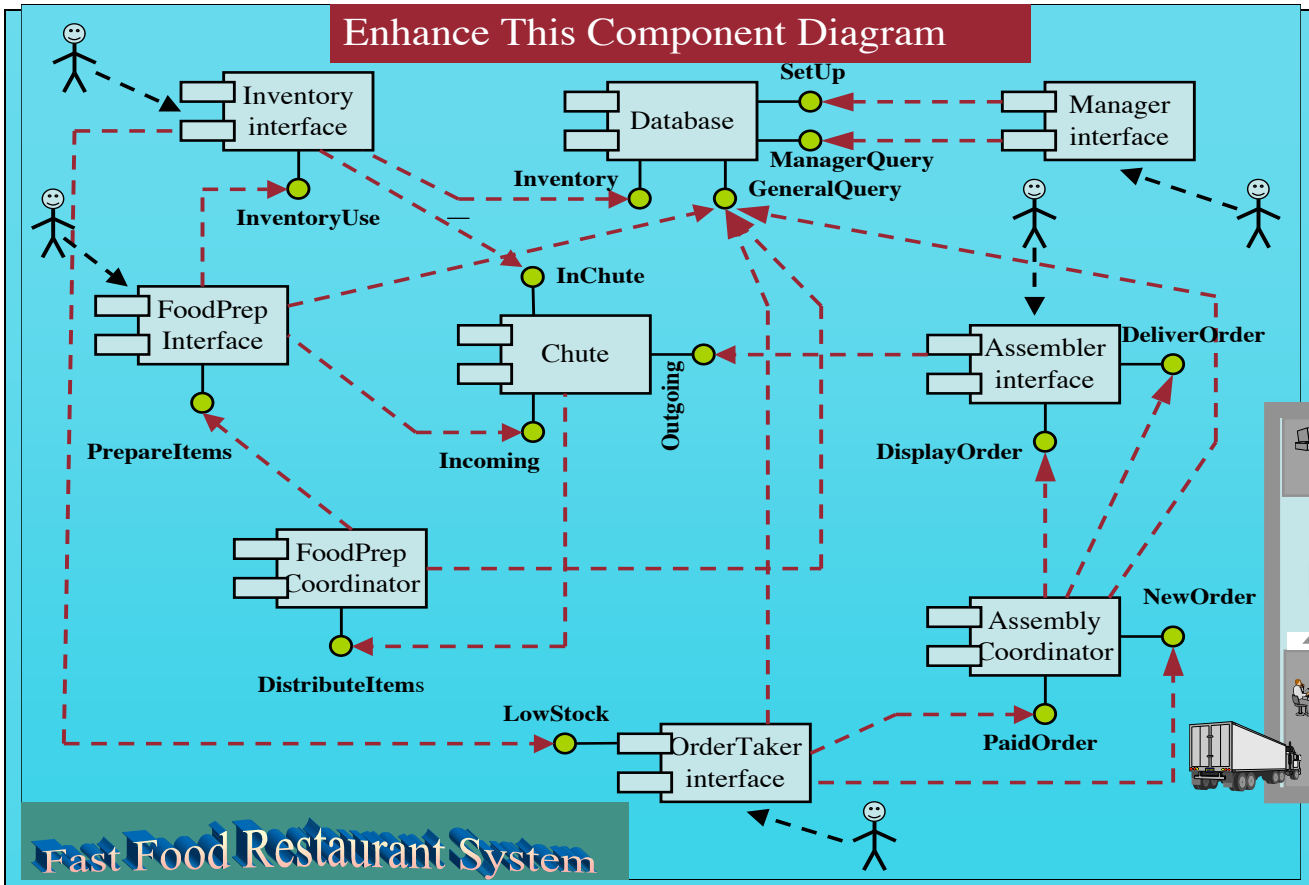
Enhance This Component Diagram

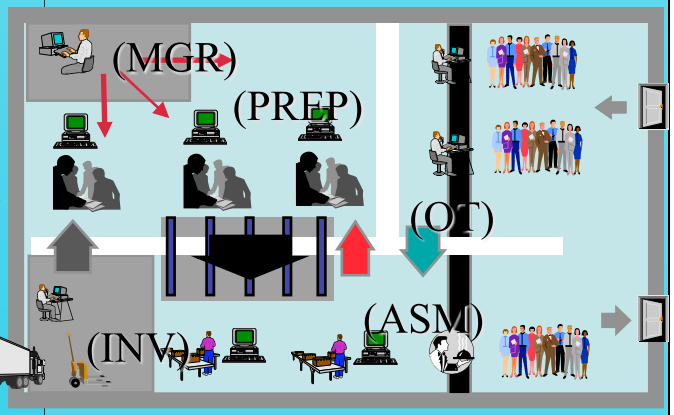


The design is based on the central repository that contains all data-structures of the restaurant such as: restaurant menu, recipes, raw materials, orders (stored, completed, paid), taxes, system configurations(i.e., number of active OT, ASM, Prep, ..).

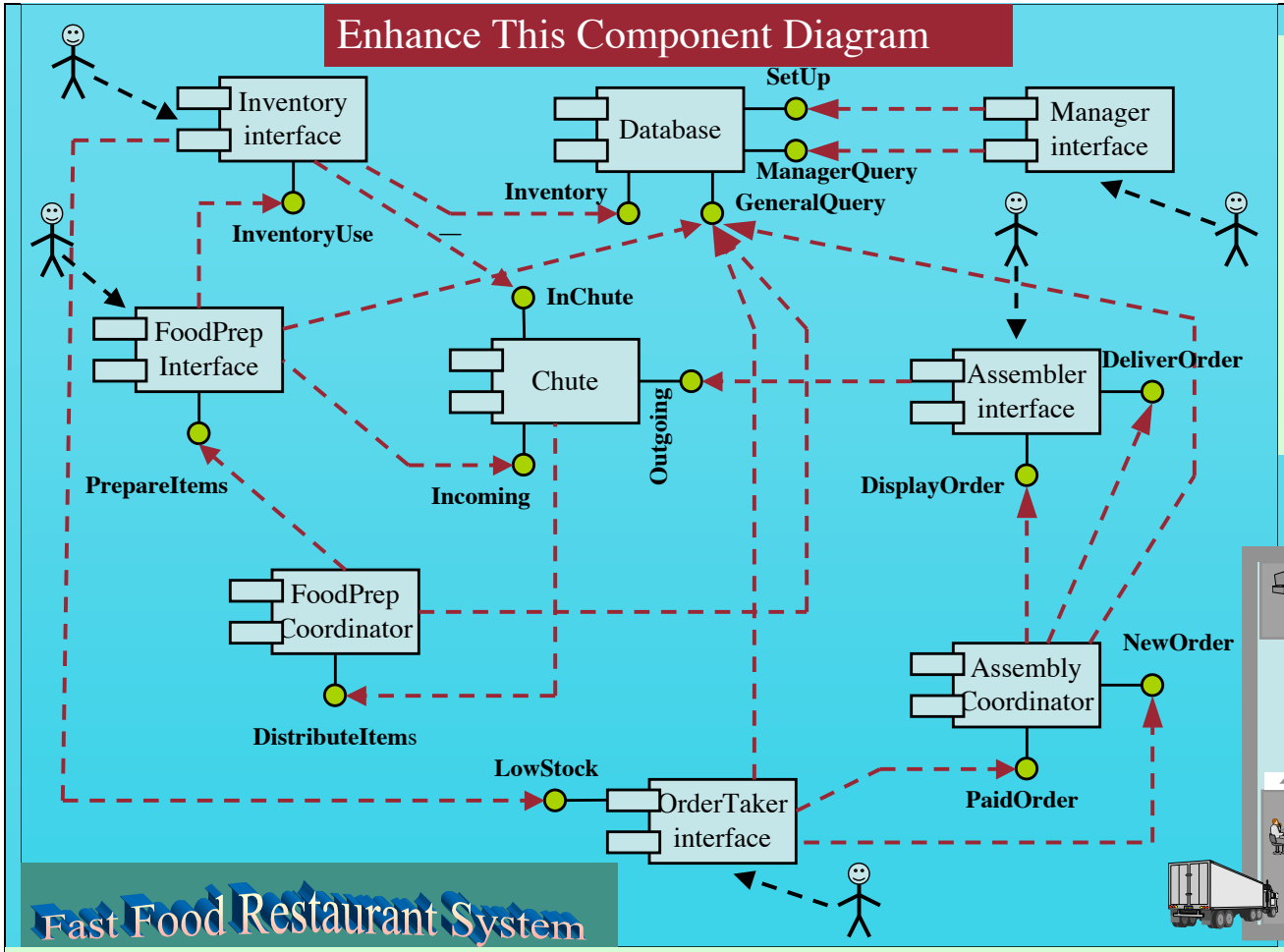


- Order taker staff interact with the OT interface to fill order for the customers.
- OT Interface retrieves restaurant menu from the DB using GeneralQuery and shows to the customer
- The incomplete/complete/paid orders are stored in the DB using GeneralQuery service.
- When an order is filled, OT INT passes the order number to the ASM COO using "NewOrder" service.
- ASM COO retrieves the order-items from DB using "GeneralQuery" . It also finds a free ASM INT and displays the order-items on that ASM INT using "DisplayOrder".
- The Assembly staff via ASM INT invokes the "Outgoing" service of the "Chute" to get the food-items from the chutes to assemble the order and put it in the box (but not deliver to customer).

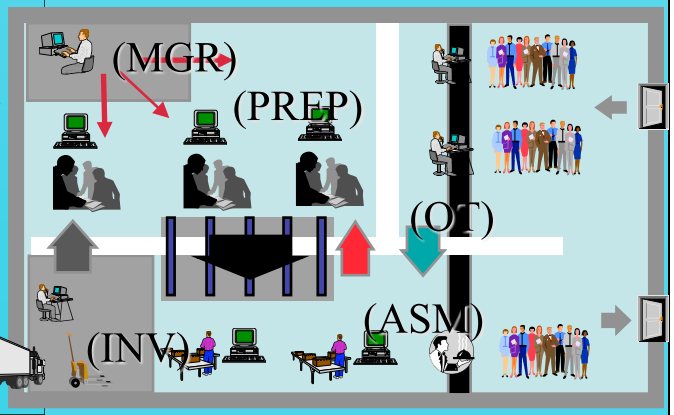




- When the customer pays for the order, OT INT informs the ASM COO using "PaidOrder", and ASM COO informs the ASM INT using "DeliverOrder". The ASM staff now deliver the order to the customer.
- When the "Outgoing" service of the Chute is invoked, Chute informs the Food COO to distribute different food-items of the order using "DistributeItems"
- Food COO obtains the information about which kitchen table prepares which food items, using "GeneralQuery" of the DB.
- Food COO assigns the food item to the proper kitchen table using "PrepareItems"
- FoodPrep INT obtains the food-item recipe from the DB interface, prepares the food-items, put them in the chutes (using "Incoming" of the Chute component) and informs the inventory INT to deduct the amount of used raw materials via the "InventoryUse" service.

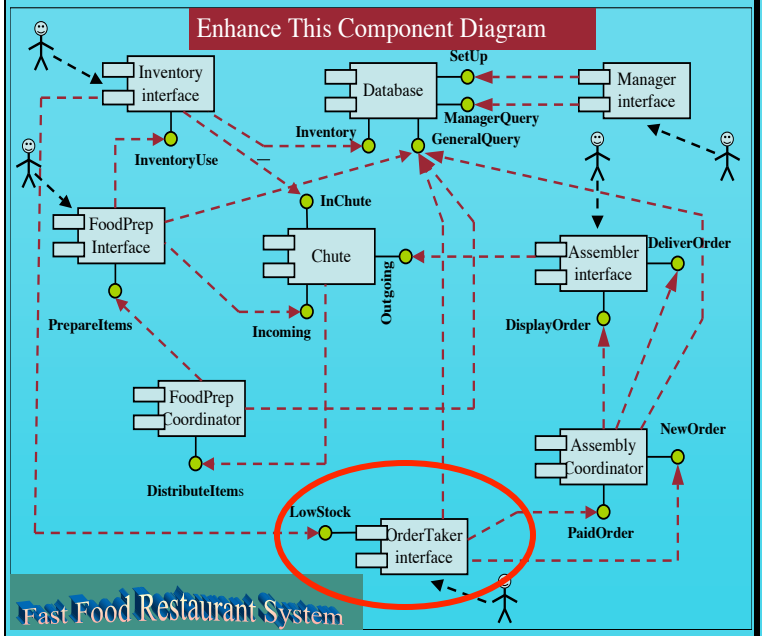


Now the task of students is to find some disadvantages of such an architecture for a restaurant system and how to improve the architecture to enhance its features and performance. A statechart will be used to show how this system works using the "guideline" on statechartswhich I posted



- INVENT INT obtains the number of exiting food items in the Chute using "InChute" in order to calculate how many food item can be prepared before the raw materials for that item(s) goes below the threshold level of that food item.
- INVENT INT retrieves and stores the amount of remaining raw-materials from the DB using "Inventory" service.
- If the level of raw material in stock goes below threshold, the INVENT INT informs the OT INT to disable that food item (using "LowStock" service) from the restaurant menu to order from suppliers.
- Manage INT uses special DB interfaces "SetUp" and " ManagerQuery" to modify the restaurant's configurations and to adjust restaurant menu and other system data structures.

Statechart for Order Taker Station



Activate Order Taker Station

/ GQ (get restaurant menu)

Order taker Station

GQ= General Query

New order initialized

Low-Stock (Item)

Standby

Item Disabled in Menu

RecallOrder / GQ (Get Stored Order)

SetupNewOrder

GoToMenu

SelectItemFromMenu / Calculate Subtotal + Tax

Item added to Order

StoreOrder / GQ (Store Order)

Order stored

CompleteOrder / GQ(store order)

PayOrder (order #)

NewOrder

PaidOrder

Overall Statechart for Restaurant System. Problem: Statechart does not correspond to the components and their interfaces