High level requirements

We need to build a canteen management application. This is an application,

## As a customer,

\* can view personal and wallet details

\* can order food items

\* can check the history of ordered food items

## As a Food Vendor,

\* can view and update (accept/deny) the ordered food items status

## Customer Login:

\* Shows list of available food items (menu) details.

\* Shows personal and Wallet details.

\* Customer can order food.

\* The order will be accepted only,when the customer had enough money for the ordered food in

His/her wallet.

\* can check the status of their delivery.

\* can check their order history.

## Food Vendor:

\* Customer can view his personal detail and order history.

\* can view the order request from customers.

\* can accept or deny the entire order.

\* customer details (custId, custName, etc..) are stored along with their wallet balance.

\* Customer can place their order in this portal. These orders go to the food vendor.

\* Food Vendor can accept/deny the order with proper comments.

\* Wallet balance should not go negative.

\* Customer can view their order status.

# Application flow

The skeletal application contains a simple screen with list of food items (Menu).

\* The "place order" button is the default button (i.e., pressing "Place order" invokes the login function).

\* Customer should enter his customer id will take the customer to "place the order screen" along with the food id he selected.

\* Customer should enter the item id and quantity and "Place Order" button to place the order

\* On existing balance amount in customer Wallet, the order should be accepted or rejected with proper error message.

\* If the order is accepted, the token number should be generate using Random functionality and the order detail should stored

in order table along with token number.

\* The token number displayed as confirmation of the order should be fetched from the database.

\* "Back to Menu" takes the customer back to the Menu list screen from all the screen.

\* Entering the "Vendor Admin Login‚Äù button in the Home screen will take to the Food Vendor login.

# Must-Have

\* Following functionalities are must to have-

1. View all food items.

2. Enter the quantity of the food item selected, order date should be generated automatically.

3. Order should be placed on adequate Wallet balance.

4. Order history will display the list of orders placed by customer.

5. Food Vendor can view the orders and accept/cancel with proper comments.

\* Functionality available via

\* Cli

\* Curl

\* Browser

\* Validations, with error messages, both on

\* the server side, to make sure the data is clean no matter what the source and,

\* the client side, to provide a responsive user interface

\* Type validations

\* Semantic validations

\* Business Logic

\* Login validation for Customer and Vendor.

\* Display history for specific Customer and specific vendor(All the status("Order Placed","Order Accepted","Order Cancelled")).

\* Display Wallet Balance for the specific customer.

\* Calculate the estimated time for food they are providing and store it in table.

\* Storing of food ordered detail will be stored in ordered table only, after getting the approval from customer

(Once after calculating the total price for the food ordered, the amount should be minus from existing wallet amount and display it to the customer (total price,wallet amount before deduction and after deduction) and get the confirmation from the Customer and store the order detail in order table and changed wallet amount in to customer table).

\* Deduction and Addition of Wallet amount depends on food ordered and cancelled.

\* Generate Random number and display as token number and should be stored in DB.

\* Vendor should accept or cancel the order based on specific reason and if it cancelled, the wallet amount should be revert back to the table in the DB and display the reverted amount (not the wallet balance)

\* On first order by the customer generate the Coupon code (FREE2020) and store it in database table, this coupon is valid only for 10 days from the date of first order

\* Customer can use this coupon(worth INR 100) only once in the given period

\* Coupon is generated only for the first order

# Nice-To-Have

\* Photos of the food items in the menu.

\* Multiple Vendors with their specialization (North Indian, South Indian, Veg, Non veg, etc..).

\* Ability to select multiple food items at once via multi-select/select-all.

\* Creating Login for the Food Vendor using password, password-reset etc.. (client and server).

\* Adding/Editing Menu details

\* Feeding money to wallet through various mode such as Net Banking/Credit/Debit cards etc.

\* Cancellation of food ordered by customer.

\* Customer can order different vendor will create a single order and internally it should create multiple orders for each product and vendor

\* Partial acceptance and deny of the order

\* Generate random coupon code on first order

# Great-To-Have

\* Confirmation alerts through SMS and Mail.

\* Sorting options on all tables

\* 'Quick filter' options on all tables.

\* Customer to re-edit their orders.

\* Ability for Customer to review and revisit their orders.

\* Different interfaces for vegetarians and non-vegetarians.

# Fantastic-To-Have

\* Rating for the food items.

\* Mobile UI

\* Conversational Interface

\* ChatBot

\* VoiceBot

\* Metrics, Monitoring, Logs, Alerts, Dashboards

\* Calories Monitoring.