

# Project: 1a1

Project Group 6 - Section 1

Project Members:

Name	Email
Khush Ketan Patel	kpatel53@ncsu.edu
Mia Glenn	mglenn2@ncsu.edu
Richa Jha	rjha3@ncsu.edu
Ishwarya Gandomsetty	igandam@ncsu.edu

## Stakeholder List

- Menu Manager — Handles menu updates, discounts, and coordinates with inventory and nutritionist.
- Inventory/ Kitchen Manager— Manages kitchen workflow by assigning orders, tracking progress, and marking orders.
- Customer — End user who explores the menu, places order.
- Delivery Manager/Driver — Handles the delivery of orders
- Nutritionist — Ensures menu item meets health standards by suggesting modifications and updating nutritional details.

## Stakeholder Details

Stockholders	Role	Bias
<u>Menu Manager</u>	Designing Menu items along with introducing nutritional labels and offers/discount	Might be more concerned towards sales growth even with the discount and offers provided to the customers as a part of customer base increase plan
<u>Inventory/Kitchen Manager</u>	Responsible for handling the inventory and the food order status	Might want more ingredients or possibly not that great quality ingredients but at a cheaper price so as to minimize the budget spent on inventory

Stockholders	Role	Bias
<u>Customers</u>	The users who will order the food via the app	Would want the good quality food but for cheaper prices
<u>Delivery Manager/Driver</u>	Pickup orders from restaurants and deliver them to customers' doorsteps	Want fair pay and reasonable delivery times but customers want fast and low-cost delivery
<u>Nutritionist</u>	Makes changes to menu items or introduces new menu items which are healthier.	Might prioritize health benefits over taste or cost.

## Use Cases

### Stakeholder : Menu Manager

▼ Use Case : Add a new Item on the menu

Goal: Add a new item to restaurant's menu

Preconditions:

- Manager is logged into system
- Necessary items details are present

Main flow:

- Manager selects the "Add a new Item" button.
- System displays a form.
- Manager enters item details along with an image.
- System validates the details.
- System saves the item and returns a success message.
- New item is added to the menu.

Sub flow:

- System validates the role of the manager.
- System compresses the image and uploads it to some cloud database.

### Alternate Flow:

- If any mandatory/invalid information is missing, systems returns the failure status with an appropriate message.
- If image upload fails, systems allows saving an image but with a warning.

### ▼ Use Case : Update a Item on the menu

Goal: Update a item to restaurant's menu

Preconditions:

- Manager is logged into system
- Item which needs to be updated is present in the system.

Main flow:

- Manager selects the items which needs to be updated.
- System displays the item information.
- Manager updates the necessary details.
- System validates the details.
- System updates the item and returns a success message.
- Updated item is shown on the menu.

Sub flow:

- System validates the role of the manager.
- If image is updated, then system compresses the image and uploads it to some cloud database.

### Alternate Flow:

- If any mandatory/invalid information is missing, systems returns the failure status with an appropriate message.
- If image upload fails, systems allows saving an image but with a warning.

### ▼ Use Case : Deleting a Item from the menu

Goal: Delete a item from restaurant's menu

Preconditions:

- Manager is logged into system
- Item which needs to be deleted is present in the system.

Main flow:

- Manager selects the items which needs to be deleted.
- System displays the item information.
- Manager clicks on the “Delete Item” button.
- System validates the details.
- System deletes the item and returns a success message.
- Deleted item is removed from the menu.

Sub flow:

- System validates the role of the manager.

Alternate Flow:

- If an item is linked to an active order, then system prevents item deletion.
- If deletion fails due to some server error, system shows appropriate error message.

#### ▼ Use Case : Apply Discounts on item/s in the menu

Goal: Applying discounts on restaurant's menu items(flat or percentage)

Preconditions:

- Manager is logged into system
- Concerned item needs to be available in the system

Main flow:

- Manager selects the items on which discount needs to be applied.
- Manager chooses the flat/percentage discount rate.
- Manager clicks on the “apply discount” button.
- System validates the details.
- System adds a new discount detail and return success message.
- Discounted rate is shown on the menu.

Sub flow:

- System validates the role of the manager.

Alternate Flow:

- If discount values exceed system-defined threshold, systems displays an appropriate message.
- If end date is before the start date, system prompts with an error message.

## **Stakeholder: Inventory/Kitchen Manager**

### ▼ Use Case : Order Received

Goal: Begin preparing food items once a new order comes in from a customer.

Preconditions:

- Customer has successfully placed and paid for an order through the system.
- Kitchen staff are available and ready to work.
- All ordered items are currently in stock.

Main flow:

- Kitchen Manager logs into the system.
- Navigates to the orders page.
- System notifies the kitchen of pending orders.
- Kitchen Manager reviews the order details
- Order gets added to the preparation queue
- Preparation begins based on order priority

Sub flow:

- Kitchen Manager assigns tasks to staff (e.g., chef makes sandwiches, prep cook makes salads)
- Kitchen Manager sets priority based on order time and customer wait time
- Kitchen Manager monitors preparation progress

### Alternate Flow:

- Kitchen is too busy with current orders: Kitchen Manager estimates longer wait time. System updates customer with realistic pickup time.
- If not kitchen staff is available, system displays an appropriate message.

### ▼ Use Case : Order Ready for Pickup

Goal: Mark order as finished and ready for pickup.

#### Preconditions:

- Order has been prepared by kitchen staff.
- All items in the order are complete and properly packaged.
- Kitchen Manager has checked the order.

#### Main flow:

- Menu manager logs into the system.
- Navigates to “Orders” page.
- Filters out orders those are prepared.
- Kitchen Manager adds special packaging for hot/cold items
- Kitchen Manager includes utensils, napkins, and condiments
- Mark them as “ready for pickup”.
- System updates their status.
- Triggers a push notification to Customer.

#### Sub flow:

- System tracks preparation times to improve future estimates

### Alternate Flow:

- System has technical problems: Kitchen Manager manually tracks ready orders. Kitchen Manager contacts customer/driver directly. Kitchen Manager updates system when it's working again.
- If the system fails to update an order, an error message appears saying “unable to update order status”.

#### ▼ Use Case : Update an Item is Out of Stock

Goal: Keep the menu up to date by removing items that are no longer available.

Preconditions:

- Kitchen Manager has access to inventory system.
- Kitchen Manager can check actual ingredient levels.
- System allows menu item status updates.

Main flow:

- Kitchen Manager identifies that an item is running low or unavailable
- Kitchen Manager marks the item as "Out of Stock" in system
- System removes or grays out the item from the customer's ordering menu
- Kitchen Manager contacts suppliers or inventory manager about restocking

Sub flow:

- Kitchen Manager estimates when item will be available again
- Kitchen Manager sets up automatic reorder for commonly used items
- System shows "temporarily unavailable" message to customers

Alternate Flow:

- If the kitchen manager searches for an item that doesn't exist, system displays an appropriate message.
- If the update fails, system displays an error message.

## Stakeholder: Customers

#### ▼ Use Case : Add an item to the cart

Goal: A customer browses available menu items and adds selected items to their shopping cart for purchase.

Preconditions:

- Customer is logged into the WolfCafe system.

- Customer has a valid account with customer role.
- Menu items are available in the system.
- Items have enough inventory.

Main flow:

- Customer navigates to the menu/items page.
- Customer browses available items and recipes.
- Customer specifies quantity desired.
- Customer clicks "Add to Cart" button.
- System adds item and quantity to customer's cart.
- System updates cart total and item count.
- System displays confirmation message.
- Customer can continue shopping or proceed to cart.

Sub flow:

- Customer can filter items by category (beverages, food, etc.).
- Customer can search for specific items by name.
- Customer views item details including price, description, and ingredients.
- System calculates running subtotal including all cart items.

Alternate Flow:

- Customer enters invalid quantity: System displays an error message and prompts customer to enter valid quantity.
- Not enough inventory for requested quantity: System will display inventory limitation message and suggest maximum available quantity.
- Item becomes unavailable after selection: System will notify customer of unavailability.
- Customer not logged in: System redirects customer to the login page.

▼ Use Case : Remove an item to the cart

**Goal:** A customer modifies their shopping cart by removing unwanted items or adjusting quantities.

**Preconditions:**

- Customer is logged into the WolfCafe system.
- Customer has items in their shopping cart.
- Customer is viewing their cart contents.

**Main flow:**

- Customer navigates to shopping cart page.
- Customer views current cart contents with items and quantities.
- Customer selects item to remove or modify.
- Customer chooses removal action (delete item or adjust quantity).
- System updates cart contents.
- System recalculates cart total.
- System displays updated cart with confirmation message.

**Sub flow:**

- Customer can see individual item prices and line totals.
- Customer can adjust quantity instead of complete removal.
- System updates subtotal, tax, and final total.
- If cart becomes empty, system displays "Cart is empty" message.

**Alternate Flow:**

- Customer brings down quantity to zero: System treats as item removal and removes item completely from cart.
- Customer increases quantity beyond available inventory: System displays inventory limitation and sets quantity to maximum available.
- Cart is empty when accessed: System displays empty cart message.
- System error during cart update: System displays error message.

#### ▼ Use Case : Place the order

**Goal:** A customer completes their purchase by placing an order with items from their cart, including payment processing and order confirmation.

Preconditions:

- Customer is logged into the WolfCafe system.
- Customer has one or more items in their cart.
- All cart items are still available and in stock.
- Customer has valid payment information.
- Sales tax rate is configured in the system.

Main flow:

- Customer navigates to checkout from their cart.
- Customer reviews order summary with all items and quantities.
- System calculates and displays subtotal.
- System applies sales tax based on configured rate.
- System displays tax amount and order total.
- Customer selects tip option.
- System calculates final total including tip.
- Customer confirms order details.
- Customer provides or confirms payment information.
- System processes payment.
- System creates order record with "Pending" status.
- System empties customer's cart.
- System displays order confirmation with order number.
- System sends order to staff for preparation.

Sub flow:

- Customer can enter custom tip amount or percentage.
- Customer can choose no tip.
- Customer can use saved payment method.
- Customer can add new payment method.
- System provides estimated preparation time.

- Staff receives notification of new order.

Alternate Flow:

- Cart is empty at checkout: System displays message to add items first.
- Sales tax rate not configured: System uses default rate or zero and logs error for admin attention.
- Customer enters invalid tip amount: System displays error message.
- Payment processing fails: System displays payment error message. Customer can retry with same or different payment method.
- Item becomes unavailable during checkout: System notifies customer of unavailable item, removes unavailable item from cart and recalculates total. Customer can proceed with remaining items or add replacements.

## **Stakeholder: Delivery Manager/Driver**

### ▼ Use Case: Pickup food

Goal: Driver arrives at WolfCafe location to collect the prepared food order for delivery.

Preconditions:

- Driver has accepted a delivery assignment.
- Order status is "Ready for Delivery".
- Driver has arrived at pickup location.
- Staff member has prepared and packaged the order.

Main Flow:

- Driver arrives at WolfCafe pickup location.
- Driver presents pickup confirmation code to staff.
- Staff verifies driver identity and assignment.
- Staff provides order package to driver.
- Driver verifies order contents against assignment details.
- Driver confirms pickup in the system.

- System updates order status to "Out for Delivery".
- System provides navigation to delivery address.

Sub flows:

- Driver checks for special handling requirements.
- Driver ensures proper packaging for transport.
- System sends notification to customer with tracking information.

Alternative flows:

- Staff cannot verify driver: Staff contacts management for verification. Driver provides additional identification.
- Order contents do not match assignment: Driver reports an error to staff. Staff corrects order or contacts customer. System updates order details if necessary.
- Order is not ready: Driver waits for order completion. System updates estimated delivery time. Customer receives delay notification.

#### ▼ Use Case: Deliver food

Goal: Driver delivers the food order to the customer at the specified delivery address.

Preconditions:

- Driver has picked up the order.
- Order status is "Out for Delivery".
- Driver has directions to delivery address.
- Customer is available at delivery location.

Main Flow:

- Driver follows direction to customer delivery address.
- Driver arrives at delivery location.
- Driver contacts customer to announce arrival.
- Customer meets driver at specified location.
- Driver verifies customer identity.

- Driver hands over food order to customer.
- Customer confirms receipt of order.
- Driver marks delivery as "Completed" in system.
- System updates order status to "Delivered".
- System processes driver compensation.

Sub flows:

- Driver calls customer using system-provided contact method.
- Driver delivery address matches order.
- System calculates delivery fee and any tips.

Alternative flows:

- Customer not available: Driver attempts to contact customer via phone or waits for reasonable time. If no response, driver contacts support. Support attempts to reach customer. If still no response, driver returns order to WolfCafe.
- Cannot locate delivery address: Driver contacts customer for clarification. Driver uses alternate address if needed. If address invalid, driver contacts support.
- Customer identity cannot be verified: Driver requests additional verification. Driver contacts support if concerns persist.
- Customer reports order issues: Driver documents customer complaint. Driver contacts support for resolution. System flags order for review.

## **Stakeholder: Nutritionist**

▼ Use Case: Suggest a new healthy menu item.

Goal: Make a request to menu manager for incorporating a new healthy dish to menu.

Preconditions:

- User is authenticated and authorized to make menu requests.
- System has existing menu items.
- Menu manager exists in the system for approval.

#### Main Flow:

- User logs into the system.
- Navigates to the Requests page.
- Fills details in the form.
- Submits the request and system stores it in "pending" state.
- Menu Manager receives the notification about the new suggestion.

#### Sub flows:

- Attaches supporting documents with the suggestion.

#### Alternative flows:

- If user tries to submit without providing some essential data, system might return an error.
- If system fails to save the data, then show an appropriate error message.

### ▼ Use Case: Ingredient Removal

Goal: Request an ingredient removal from existing menu item.

#### Preconditions:

- User is authenticated and authorized to make menu requests.
- System has existing menu items.
- Menu manager exists in the system for approval.

#### Main Flow:

- User logs into the system.
- Navigates to requests page.
- Selects an existing menu item.
- Clicks on item removal and chooses what ingredients to remove.
- Provides some comments and submits the request.
- System stores the request and marks it as pending for approval.
- Menu Manager receives a notification.

#### Sub flows:

- Attaches supporting documents with the suggestion.

Alternative flows:

- If user tries to submit without providing some essential data, system might return an error.
- If system fails to save the data, then show an appropriate error message.

## Prompt Crafting

**Zero-shot prompting:** A method of engaging a large language model (LLM) where you ask it to complete a task without giving it any examples of the expected result. This technique banks on the model's existing knowledge from its initial training to deduce the correct answer.

Example: For the purpose of designing the initial look of our application landing page, we used the help of chatgpt to create a design image.

Below is the prompt given:

*Create a website page design image with the below details  
 1st - Landing Page  
 - The name of the cafe - 'WolfCafe' and the login form with the login button  
 below the form - where there is an option for entering the username  
 'USERNAME' and password 'PASSWORD' in the form and option to select the  
 role with which you want to login - menu manager , kitchen/inventory manager,  
 customers - 'Choose your role'*

*Below is the design generated:*



# WolfCafe

## Login

USERNAME

PASSWORD

Choose your role

- Menu Manager
- Kitchen/Inventory Manager
- Customers

LOGIN

**Careful prompting (few shot prompting):** Involves providing the model with a few instances of the desired output. This guidance helps steer the model toward a more precise and consistent response, proving especially useful when a particular format or complex reasoning is required.

Example: For the purpose of designing the initial look of our application landing page, we used the help of chatgpt to create a design image but this time we used few shot prompting to design it better to our requirement alignment

This is the prompt given: *Create a website page design image where The theme of the design should be red, white and black with the below details*

*1st - Landing Page - The name of the cafe - 'WolfCafe' (with a howling wolf in a circular border image as the logo image)*

*and the login form with the login button below the form - where there is an option for*

- entering the username 'USERNAME' , with small reference image*
- password 'PASSWORD', with small reference image*
- option to select the role with which you want to login 'CHOOSE YOUR ROLE: '*
- menu manager , kitchen/inventory manager, customers, delivery manager...with a small reference image for each of the options option menu manager, kitchen/inventory manager, customer, delivery manager*

*Below is the design generated:*



# WolfCafe



USERNAME



PASSWORD

## CHOOSE YOUR ROLE:



Menu Manager



Kitchen/Inventory Manager



Customers



Delivery Manager

**LOGIN**