Stakeholders

Primary: Groups who will be directly interacting with the application and have the greatest direct influence on the development

- Customers
- Delivery Drivers & Couriers
- Restaurant Staff (Managers, Kitchen, Front End)
- Platform Staff (Admin, Developers, Support, Finance, Marketing)

Secondary: Groups who may not be directly interacting with the application, but still have some influence on the development (directly or indirectly)

- Health Department/Regulators
- First Time/External Users (People who don't use the platform yet)
- Suppliers
- Third Party Service Providers (ex., Payment, Maps, Cloud Hosting, SMS/Email)
- Competitors
- Insurance Providers

Stakeholders' Biases

Platform Staff

Irrelevant Stakeholders

- Competitors don't want this platform to succeed, so they're desires are irrelevant.

Stakeholders that Clash

- Third Party Service Providers will aim to maximize their own profits, while the Platform staff will need to get costs down while still offering the features provided by Third Party Service Providers (ex., Maps, Online Payment)
- Restaurants will want to maximize the revenue from food sales, while the platform will also want its share of the profit, as the restaurant utilizes its platform
- Customers want flashy features, reliability, and an intuitive design. Platform staff want a system that is stable and has an efficient backend, and clean code.

Restaurant Staff

Irrelevant Stakeholders

- Insurance providers do not play a direct role in the Restaurant Staff's job, and as such, the influence of the insurance providers should play no role in the Restaurant Staff's perspective of the application.
- The Customer Interface plays no role in the Restaurant Interface; as such, any customer concerns on the interface should not impact the Restaurant Staff's interaction with the application. Additionally, the Restaurant Staff have different information on their interface, which is irrelevant to the Customer

Stakeholders that Clash

- Customers want the cheapest option, while Restaurants want to turn a profit and maximize their own revenue
- Restaurant Management wants to maximize the profits of the business, which conflicts with the average staff member who wants to work in a low-stress environment.

Delivery Drivers & Couriers

Stakeholders that Clash

- Customers want fast delivery regardless of any factors, including weather or traffic. Drivers & Couriers want to be able to work in safe environments, even if that means there is a delay in stormy weather, or if encounter traffic jams
- Insurance Providers want to minimize the risk of Insurance Claims while maximizing profits. Drivers & Couriers want to be financially protected in case of any accidents or emergencies that occur during a delivery or pickup.

Zero-Shot Prompting & Careful Prompting

Zero-shot prompting is asking a model to complete a task or answer a question without giving it any prior knowledge. This is useful when the task is simple enough for the model to easily understand and perform, or if you're in need of a quick answer to a question. This method is limited by the model's lack of context, which can lead to it missing out on some aspects of the prompt.

Careful prompting is when you carefully choose wording, details, context, and examples for the model to use when answering. This is useful when the task is more complex or a specific output format is required. Unlike zero-shot prompting, however, careful prompting takes much longer and requires more work on the input side in order to receive the correct output.

Use Cases

UC1: Order a Drink as a Customer
Preconditions:
• The customer is logged in.
• At least one menu item is available.

Main Flow:
The user opens the application, triggering the system to **[Display Menu]**. The user then makes a selection, which executes the **[Select Item]** subflow. The user provides final confirmation to **[Place Order]**, prompting the system to **[Confirm Order]** and finalize the transaction.

Subflows:
• **[Display Menu]** System retrieves and displays available items.
• **[Select Item]** Customer chooses an item from the list.
• **[Place Order]** Customer confirms (e.g., taps "Order Now").
• **[Confirm Order]** System creates an order number, sets status to ORDERED, and shows a confirmation message.

Alternative Flows and Edge Cases:
• [AF1: No Available Items] At **[Display Menu]**, if no items are available, show: "Sorry, no items are currently available." and return to start.
• [AF2: Item Unavailable After Selection] At **[Place Order]**, if the item is no longer available, show: "Sorry, that item just sold out," and return to **[Display Menu]**.

UC2: View and Fulfill a New Order
Preconditions:
• The Staff member is logged into the Staff Dashboard with the STAFF role.

Main Flow:

The staff member opens the dashboard, prompting the system to **[Check Orders]**. The staff selects an order to **[Prepare Order]** and, upon completion, chooses to **[Mark Fulfilled],** which directs the system to **[Update Status]** and remove the order from the active list.

Subflows:

- **[Check Orders]** System retrieves all orders with status ORDERED.
  – If none exist, show "No new orders" [No Orders].
- **[Prepare Order]** Staff prepares selected order.
  – If unable to complete, go to [Cancel Order].
- **[Mark Fulfilled]** Staff selects "Mark as Fulfilled"; system sets status to FULFILLED.
- **[Update Status]** Order is removed from the ORDERED list.
- **[Cancel Order]** Staff selects "Cancel Order"; system sets status to CANCELLED.

Alternative Flows:

- [AF1: No Orders] At **[Check Orders]**, if no ORDERED orders exist, the System shows "No new orders."
- [AF2: Missing Ingredients] At [Prepare Order], if the order cannot be completed, go to **[Cancel Order]**.

UC3: Delivery Driver Delivers Order
Preconditions:

The User must be Logged In, and their account must have the *Delivery Driver/Courier* permissions [AF1].

Mainflow:

On the main home page, the User selects **"Check Orders," which displays** all orders assigned to the User that are ready for pickup [AF2]. The User is then able to select an Order, then select the ***Pickup*** option [AF3][AF4]. The User is then directed to the Map, which updates to show the most efficient and safest route for the Driver to take to deliver the order to the Customer [AF5]. Once the Driver delivers the Order, they select the ***Delivered*** option, which will remove the Order from the **Active Orders** and add it to the **Delivered Orders** under the Customer's account [AF6].

Subflows:

- ***Pickup***: Once selected, the Order is removed from the **Orders Ready** list (as seen under **Check Orders**), and is moved to the **Active Orders** list under the Users account. Additionally, the Customer receives a notification that the Order has been picked up and will be delivered soon.
- **Map**: Displays the map of where the User is, as well as the route that should be taken, as well as any potential alternative routes. The Map page will automatically update based on traffic and weather conditions to provide the safest and quickest route.

Alternative Flows:

- [AF1]: If the User does not have **Delivery Driver/Courier** permissions, they will be redirected to the Customer view of the application, but will be able to go under the Support section of the application to request assistance
- [AF2]: If no order is available and marked as ready for pickup, the User is unable to select ***Pickup***
- [AF3]: If more than one order is marked as ready, all orders can be selected and the ***Bulk Pickup*** option can be selected

UC4: Restaurant Staff Create Menu Item
Preconditions:

The User must be Logged In, and their account must have the *Restaurant Staff* permissions [AF1].

Main Flow:
On the main home page, the User selects the Menu option, which will display the list of all items that are publicly available for purchase on the application [AF2]. The User then selects the ***Add Item*** option, which will redirect them to the Item Creation page. Following the prompts on the Item Creation page, the User will then add Item details consistent with the configuration set by the Restaurant Administrator [AF3][AF4]. Once all details have been added, the User will click the ***Finalize*** button, which will add the Item to the Menu, while returning the User to the Menu page [AF5].

Subflows:

- ***Add Item***: When selected, a template Item object is created, which is either filled through the Item Creation page or is deleted if the User exits before finalizing.
- **Item Creation**: Displays the fields for the Item Creation, which are either required or optional as set by the Restaurant Administrator.

Alternative Flows:

- [AF1]: If the User does not have *Restaurant Staff* permissions, they will be directed to the Customer view of the application.
- [AF2]: If there are no items on the **Menu**, the only option available to the User will be to ***add an Item***.
- [AF3]: If the User inputs an Item Name that is identical to another active Item, a prompt will appear stating that the name cannot be the same and should be changed.
- [AF4]: Restaurant Administrators must designate at least one field as required for **Item Creation**.
- [AF5]: If a User attempts to ***Finalize*** an Item without completing each required field, a prompt will appear showing which fields need to be completed.

UC5: Scheduling Food Orders

Preconditions:
The User must be logged into the application to schedule orders ahead of time [AF1]. The User must have selected items from the restaurant's menu and chosen a specific date/time window for pickup or delivery [AF2]. A valid payment method must be provided before the scheduled order can be confirmed [AF3].

Main Flow:
On the application home page, the User selects the restaurant and views the available menu items [AF4]. The User then selects the items they want to order and proceeds to the scheduling option. The User selects their preferred date and time for order pickup or delivery [AF5]. The User pays for the scheduled order or selects the "Pay at Store" option if available. Once confirmed, the system validates the time slot and creates the order in the restaurant's system [AF6]. The User then receives a confirmation message with order details and tracking information [AF7].

Subflows:

- **Modifying Scheduled Order:** If the User wishes to edit the date or time of a scheduled order, they must do so before the delivery process has begun [AF8]. The User can update the scheduled order either through the application or by contacting the restaurant directly. Once confirmed, the system updates the order record.

- **Canceling Scheduled Order:** If the User cancels the scheduled order before the designated pickup/delivery time, the system refunds the payment (if already made) and sends a cancellation confirmation to the User [AF9].

Alternative Flows:

- [AF1] If the User is not logged into the application, they will be prompted to log in before scheduling an order.
- [AF2] If the User attempts to schedule an order without selecting items, they will be redirected to the menu to make a selection.
- [AF3] If the payment method cannot be validated, the system prevents the order from being placed and prompts the User to update payment details.
- [AF4] If the selected restaurant is not available or has no items, the system will display a message stating that no orders can be placed.

UC6: Inventory Management of Items

Preconditions:
The restaurant has an inventory system that is specific to the system [AF1]. All menu items are registered in the system [AF2]. Administration, staff teams, and vendors must have a login to the system and authorization to view it [AF3].

Main Flow:
Administration navigates to the inventory management system [AF4]. The system displays all the items, all the stock levels, and when to restock. The Admin can select an item to update or add a new item to the inventory that needs to be shipped. The Admin can update the stock quantity and reflect the updated stock levels in real time as well [AF5]. The Admin can also contact vendors through the application if restocking is needed. The system confirms the updates, and the changes go through it [AF6]. The system updates the application customers use, in case something is out of stock [AF7]. Finally, the Admin logs out of the management system.

Subflows:

- **Inventory Reports:** Admin can initiate this to make sure the management system is running and updating correctly [AF8].
- **Low Stock Alerts:** The system checks if stocks are below a certain point. The system generates a notification to management for ordering more. The system may also automatically put an order request in for restocking [AF9].
- **Inventory Deduction After Order:** When customers place an order, the system automatically deducts the item from inventory. If the item is unavailable, the system automatically updates the customer application to show that the item is unavailable [AF10].

Alternative Flows:

- **[AF1]** If the restaurant does not have an inventory system specific to the application, the Admin cannot manage inventory.

- **[AF2]** If a selected item does not exist in the menu database, the system shows an error message.
- **[AF3]** If a User without authorization attempts to access inventory management, the system denies access.
- **[AF4]** If the database fails due to technical issues, the system cannot update inventory, displays an error, and logs the failure.

UC7: Apply Discounts/Promotions

Preconditions:
The Customer must be logged into the system to apply their discount through their portal [AF1]. The Customer must have selected items for which the discount is approved [AF2]. The promotion or discount must exist in the system and be validated [AF3].

Main Flow:
The Customer navigates to check out after selecting items or checking out online [AF4]. The app displays the available discounts or promo codes. The Customer applies the discount by either selecting the promotion, entering the promo code manually, or having the discount on paper [AF5]. The system validates the discount once scanned at the register or online. If validated, the system recalculates the order total with the discount applied [AF6]. The system then saves the order with the applied discount and continues to checkout [AF7].

Subflows:

- **Automatic Promotions:** The system automatically applies the discount once checkout is finished [AF8].
- **Adding More Than One Discount:** The system checks whether more than one discount is added and either validates it or rejects it [AF9].

Alternative Flows:

- [AF1] If the Customer is not logged into the system, they will be prompted to log in before applying a discount.
- [AF2] If the discount does not apply to the selected items, the system will not allow it to be used.
- [AF3] If the promotion has expired, is invalid, or has already been used, the system rejects it and displays an error message.

UC8: Admin Menu Deletion

Preconditions:
The user is logged in with the ADMIN role. The item to be deleted exists in the system.

Main Flow:
The ADMIN navigates to the **Item Management** section of the dashboard.The system triggers **[Fetch Items]** to load the list of items. The admin locates the item to be deleted and clicks the **“Delete”** button. The system prompts for confirmation (e.g., modal dialog) [AF1]. The admin confirms the deletion request. The system checks if the item is associated with any active orders ([Check Order Association]) [AF2]. The system proceeds to **[Delete Item]** [AF3]. The system confirms success and updates the item list.

Subflows:

- **[Fetch Items]:** The system retrieves and displays all existing items in the database.
- **[Check Order Association]:** The system checks if the selected item is referenced in any orders with a status.
- **[Delete Item]:** The system removes the item from the catalog.

Alternative Flows:

- [AF1] If the admin clicks “Cancel” or closes the dialog, the system exits the deletion process. No changes are made.
- [AF2] At [Check Order Association], if the item is found in any pending orders, the system blocks deletion and shows “Cannot delete item. It is associated with one or more active orders.”
- [AF3] If Database Error occurs during deletion, the system logs the error and shows “An error occurred while deleting the item. Please try again later.”

UC9: Register as Admin, Staff, or Customer

Preconditions:
The user is not logged in. The WolfCafe registration page is accessible.

Main Flow:
The User navigates to the **Registration Page** and fills in required details - Full Name, Email Address, Password, Confirm Password, Select Role (Admin, Staff, or Customer). The user clicks “Register”. The system validates the input fields [AF1] and checks if the email is already registered [AF2]. If the role selected is **Admin** or **Staff**, the system verifies user authorization [AF3]. The system creates the user account with the selected role. The system sends a confirmation email with an activation link. The user confirms the email by clicking the activation link. The system activates the user account.

Subflows:

- **[Input Validation]:** The system validates email format, password strength, password confirmation, and required fields.
- **[Authorization Check for Admin/Staff]:** The system checks for authorization tokens or invite codes for Admin/Staff roles.
- **[Send Confirmation Email]:** The system generates and sends activation email to the user's email address.
- **[Account Activation:]:** Upon clicking the activation link, the system activates the user account.

Alternative Flows:

- [AF1] If any required field is missing or invalid (e.g., invalid email format, password mismatch), the system shows specific error messages and the user is prompted to correct errors.
- [AF2] If the email exists, the system shows “Email is already registered. Please login or use another email.”
- [AF3] If the user lacks required authorization (e.g., no invite code), the system rejects registration with message - “Registration as Admin or Staff requires authorization. Contact system administrator.”

UC10: View Order History (Customer/Staff/Admin)

Preconditions:
The user is logged in with a valid role (**Customer**, **Staff**, or **Admin**). At least one order exists in the system (globally or for the user).

Main Flow:
The user navigates to the **Order History** section from their dashboard. The system displays the **order history interface**, showing a table of past orders [AF1]. The user (optionally) applies filters like - Date range, Status, Order Id. The system processes filters and fetches relevant orders[AF2]. The user views a list of past orders based on role. The user selects an order to **view details** and the system displays the detailed view.

Subflows:

- **[Filter Orders]:** The user enters optional filter criteria.
- **[Fetch Orders by Role]:** The system retrieves the logged-in customer's orders, if the user is a Customer and all orders, if the user is Staff or Admin
- **[View Order Details]:** On selecting an order, the system shows a full order breakdown.

Alternative Flows:

- [AF1] If the system fails to fetch orders from the backend, the error message ‘Unable to load order history. Please try again later." is displayed.
- [AF2] At **[Filter Orders],** if the query returns no results, the system shows “No orders found for selected filters.”