Use Cases

UC 1: Add a New Item to Inventory

Pre-reqs	Short Description	Main Flow	Alternative Flows
The user (Staff or Admin) is logged into the system. The user has the appropriate role-based access (Staff or Admin).	Staff or admin can add a new item to inventory	1.User logs into system 2. User clicks “Add Item” button 3. User fills out the required information including name, description, and amount 4. User clicks the “Submit” button [Invalid Input] [Duplicate Name] 5. The new item is added to the inventory	**[Invalid Input]**: One or more input fields are empty **[Duplicate Name]**: Name of new item already exists in inventory

UC 2: Delete an Item in Inventory

Pre-reqs	Short Description	Main Flow	Alternative Flows
The user (Staff or Admin) is logged into the system. The user has the appropriate role-based access (Staff or	Staff or admin can add an item in inventory	1.User logs into system 2. User clicks “Delete Item” button 3. User searches for the name of item to be deleted in dropdown menu and selects it 4. User clicks the “Submit” button [Cannot Delete] 5. The item is removed from inventory	**[Cannot Delete]**: Item to be delete exists in at least one recipe and cannot be deleted

Admin).			

UC 3: Create A Recipe

Pre-reqs	Short Description	Main Flow	Alternative Flows
The user (Staff or Admin) is logged into the system. The user has the appropriate role-based access (Staff or Admin).	Staff or admin can create a new recipe from items in inventory	1.User logs into system 2. User clicks “Create Recipe” button 3. User fills out the required information including name, description, items from inventory, and cost 4. User clicks the “Submit” button [Invalid Input] [Duplicate Name] 5. The new recipe is added to the menu	**[Invalid Input]**: One or more input fields are empty or have an invalid value **[Duplicate Name]**: Name of new recipe already exists on the menu

UC 4: Delete a Recipe

Pre-reqs	Short Description	Main Flow	Alternative Flows
The user (Staff or Admin) is logged into the system. The user has the appropriate role-based access (Staff or Admin).	Staff or admin can delete a recipe from the menu	1.User logs into system 2. User clicks “Delete Recipe” button 3. User searches for the recipe name to be deleted in dropdown menu and selects it 4. User clicks the “Submit” button [Cannot Delete] 5. The recipe is removed from the menu	**[Cannot Delete]**: Recipe has a current order in line for a customer and cannot be deleted until order is fulfilled

UC 5: Edit a Recipe

Pre-reqs	Short Description	Main Flow	Alternative Flows
The user (Staff or Admin) is logged into the system. The user has the appropriate role-based access (Staff or Admin).	Staff or admin can edit an recipe from the menu	1.User logs into system 2. User clicks “Edit Recipe” button 3. User searches for the recipe name to be edited in dropdown menu and selects it 4. User edits parameters in question such as name, item counts, price, etc. 4. User clicks the “Submit” button [Invalid Input] [Duplicate Name] 5. The recipe is updated in the menu	**[Invalid Input]**: One or more input fields are empty or have an invalid value **[Duplicate Name]**: Name of edited recipe already exists on the menu

UC 6: Manage Staff & Customers

Pre-reqs	Short Description	Main Flow	Alternative Flows
The user (Admin) is logged into the system. The user has the appropriate role-based access (Admin).	The admin needs to be able to create/edit/ delete staff/customer user accounts	1. An admin selects the "Manage Staff and Customers" option. 2. The system displays a list of all existing staff and customer users. 3. The admin can choose to: a. Create a New User: i. The admin selects the "Add New User" option. ii. The system prompts the admin to enter details (e.g., username, password, role, contact information). [Invalid Input] iii. The admin enters the	[Invalid Input]: If the admin enters invalid information (e.g., missing required fields) while creating or editing a user, the system displays an error message indicating the required fields and prompts the admin to correct the input.

		information and submits it. iv. The system validates the information and saves the new user, displaying a confirmation message. b. Edit an Existing User: i. The admin selects a user from the list to edit. ii. The system displays the current details of the selected user. iii. The admin modifies the necessary fields and submits the changes. [Invalid Input] iv. The system displays a confirmation message. c. Delete a User:	

UC 7: Customer Orders

Pre-reqs	Short Description	Main Flow	Alternative Flows
The user (Customer) is logged into the system. The user has the appropriate role-based access (Customer).	Customers need to be able to add items to their order and then make a purchase while leaving a tip.	1. A customer selects the Order Items option from the menu. 2. The system displays a list of available recipes/items. 3. The customer browses the items and selects one or more recipes/items to add to their order. 4. The customer can specify the quantity for each selected item. 5. The system calculates the order	[Order Time Limit]: If the customer takes too long to finalize the order, the system times out and prompts them to restart the ordering process.

		total, including the selected items, applicable sales tax, and displays it to the customer. 6. The customer is prompted to enter a tip, choosing from options of 15%, 20%, 25%, or a custom amount. 7. The customer reviews the order summary and confirms the order. [Order time out] 8. The system processes the order and displays a confirmation message with the order details. 9. The customer is returned to the order menu or home screen.	

UC 8: View & Fulfill Orders

Pre-reqs	Short Description	Main Flow	Alternative Flows
Staff is logged into the system.	Staff users need to be able to view orders and fulfill them.	1. Staff can look at current orders waiting to be fulfilled. 2. Staff members can choose which order they would like to fulfill. 3. Staff member chooses to fulfill an order [Order	[Order fulfilled by other Staff]: If when a staff member tries to fulfill an order the order has already been fulfilled by another staff member the staff member is displayed an error message stating that the order has already been fulfilled and the page is reloaded to show current orders waiting to be filled.

		fulfilled by other Staff]. 4. The system updates the order to show that it is fulfilled. 5. The person who ordered is notified that their order has been fulfilled by a staff member and is ready for pick up [UC 5].	

UC 9: Pickup Notification

Actors	Short Description	Main Flow	Alternative Flows
Customers and staff members are logged into the system. Customer places an order [UC #3] and a staff member fulfills that order [UC #4].	Customers need to be able to see when their order is fulfilled so they can pick it up.	1. Customer's screen displays a notification saying that their order is ready for pickup. 2. The system logs information and updates the customer's view of their order even after the notification goes away. 3. Customer picks up their order and confirms in the system that they received it [No Confirmation].	[No Confirmation]: If the customer fails to complete the confirmation of pickup, then after 2 hours, the system will log the order as confirmed anyways since the customer did not pick up their order in time.

UC 10: Set Sales Tax Rate

Pre-reqs	Short Description	Main Flow	Alternative Flows
Admin is logged into the system.	Admin users need to be able to set a sales tax rate for the system.	1. Admin selects the option to set the sales tax rate. 2. Admin enters the new sales tax rate. 3. The system validates the sales tax rate [Invalid Rate]. 4. The system updates the sales tax rate and logs the update with Admin details. 5. Admin is returned to the main menu.	[Invalid Rate]: If the input sales tax rate is not a valid positive decimal number (ex. greater than or equal to 0 and less than or equal to 100), an error message is displayed, and the Admin is returned to the form to enter a valid rate. The tax rate isn't updated.

UC 11: View Inventory

Pre-reqs	Short Description	Main Flow	Alternative Flows
Staff/Admin is logged into the system.	Staff or Admin can view stock levels in report format.	1. User selects “Inventory Report.” [No Data] 2. System compiles current stock quantities. 3. Report is displayed and available for download.	[No Data]: If inventory is empty, show “No items available.”

Justification, Reflection, and Link

Here is the link to the conversation that was used to make these use cases and reasoning for cutting out others: https://copilot.microsoft.com/shares/mNsxeiCEzopJnnNx4KxhU

When brainstorming with the help of LLMs, our list of potential use cases grew well beyond the project requirements. This quickly expanded the scope past what was feasible for an MVP. To cut down on the number of use cases we had, we returned to the official

requirements and ensured that each one was covered by at least one use case. Any use case that did not directly map to a requirement was removed. For example, features like loyalty points and advanced analytics for admins were attractive but not required. These were cut because they would have added complexity. In short, we prioritized coverage of the core functionality over extra convenience or “nice to have” features. This aligns with the principle of a minimum viable product, where we should simply have requirements here that we got directly from the people we are making the product for. While there are aspects that we would like to add on, these are not what was explicitly asked for, and are therefore unnecessary.

Negative impacts or disappointments for stakeholders

Cutting certain use cases inevitably disappointed some stakeholders:

Customers may be disappointed that features like custom meal suggestions or saved favorite orders are not available in the MVP. While the app still allows them to place and pick up orders, some of the polish and personalization is missing.

Staff may feel frustrated that more advanced queue management features are absent, which could help them balance workloads during peak times.

Admins might have expected more detailed reporting or monitoring tools to demonstrate profitability and efficiency, but the MVP focuses only on enabling orders to flow correctly.

Future Developers may find that without certain long-term features in place, the codebase looks too narrowly designed, meaning they’ll have to re-architect for scalability later.

Changes Made to Appease stakeholders

To appease some stakeholders, we decided to add a couple of extra use cases: the ability to delete recipes and to delete items from the inventory. We included these because they solve practical problems for our different stakeholders. For customers, removing outdated recipes keeps the menu clear. For staff, being able to take ingredients out of the inventory makes daily work smoother and reduces mistakes when preparing orders. For admins, these features provide better control over keeping the system accurate and reliable. For future developers, having delete functionality built in from the start prevents the need to tack it on later. These small additions strengthen the MVP by making it more realistic and maintainable, without stretching the project beyond its intended scope. We believe that the stakeholders would prefer a fully functional, completely fleshed out product that does a few things extremely well, rather than a sprawling product that has many features but they are not executed well, or documented well which is the case of having non-written down use cases.