

## New Use Cases

### Use Case 1: A Customer Places an Order

#### UC 1.1 Preconditions

The customer has accessed the ordering app and is logged in or continuing as a guest.

#### UC 1.2 Main Flow

The customer browses the restaurant menu. [S1]

The customer adds one or more items to their cart. [S2]

The customer proceeds to checkout. [S3]

The customer confirms their delivery address and submits the order. [S4] [A1]

#### UC 1.3 Subflows

[S1] The customer taps the “Menu” tab.

[S2] The customer clicks “Add to Cart” under a meal option.

[S3] The customer clicks “Checkout” and fills in address and contact details.

[S4] The customer presses “Place Order,” and the system creates an order record.

#### UC 1.4 Alternative Flows

[A1] If the restaurant is closed or the item is unavailable, the customer is shown a message: “This item cannot be ordered right now.”

---

### Use Case 2: A Customer Makes a Secure Payment

#### UC 2.1 Preconditions

The customer has items in their cart and is on the checkout screen.

#### UC 2.2 Main Flow

The customer selects a payment method (credit/debit card, wallet). [S1]

The payment is processed through the payment gateway. [S2]

The system confirms a successful transaction. [S3] [A1]

#### UC 2.3 Subflows

[S1] The customer taps “Pay Now.”

[S2] The payment gateway validates the transaction.

[S3] A confirmation screen appears: “Payment successful, order placed.”

### **UC 2.4 Alternative Flows**

[A1] If the transaction fails (insufficient funds, network issue), the system displays: “Payment could not be processed. Please try again or use another method.”

---

## **Use Case 3: Restaurant Staff Receive a New Order**

### **UC 3.1 Preconditions**

A customer has successfully placed and paid for an order.

### **UC 3.2 Main Flow**

The restaurant system receives the new order. [S1]

The order details appear on the kitchen display. [S2]

The staff begin preparing the meal. [S3] [A1]

### **UC 3.3 Subflows**

[S1] The restaurant tablet beeps with a notification: “New Order.”

[S2] The staff click into the order to view details (items, notes, timing).

[S3] The chef starts preparation.

### **UC 3.4 Alternative Flows**

[A1] If the restaurant is too busy, the manager can reject the order. The system notifies the customer and issues a refund.

---

## **Use Case 4: Restaurant Manager Updates Menu Items**

### **UC 4.1 Preconditions**

The restaurant manager is logged into their admin account.

### **UC 4.2 Main Flow**

The manager enters the menu management section. [S1]

The manager updates item details such as price, description, or availability. [S2]

The system saves changes and updates the menu for customers. [S3] [A1]

### **UC 4.3 Subflows**

[S1] The manager clicks “Manage Menu.”

[S2] The manager edits “Grilled Chicken Sandwich” to update its price from \$8.99 to \$9.49.

[S3] The system refreshes the item on the live menu.

#### **UC 4.4 Alternative Flows**

[A1] If invalid data is entered (e.g., “nine dollars”), the system prompts: “Please enter a valid price in decimal format.”

---

### **Use Case 5: Delivery Driver Accepts a Delivery Request**

#### **UC 5.1 Preconditions**

A customer order has been accepted by a restaurant. A driver is logged in and marked “Available.”

#### **UC 5.2 Main Flow**

The system sends a delivery request to the driver. [S1]  
The driver views order details (pickup and drop-off). [S2]  
The driver accepts the task. [S3] [A1]

#### **UC 5.3 Subflows**

[S1] The driver’s phone buzzes with a notification.  
[S2] The driver opens the request to see earnings, restaurant location, and customer address.  
[S3] The driver clicks “Accept.” The order is now assigned to them.

#### **UC 5.4 Alternative Flows**

[A1] If the driver ignores or rejects the order, the system reassigns it to another driver.

---

### **Use Case 6: Customer Tracks Order Status**

#### **UC 6.1 Preconditions**

The customer has successfully placed an order.

#### **UC 6.2 Main Flow**

The customer opens the “My Orders” section. [S1]  
The system shows the current status of the order (e.g., accepted, in preparation, out for delivery). [S2]  
The system updates the status as it changes. [S3] [A1]

#### **UC 6.3 Subflows**

[S1] The customer taps “Track Order.”  
[S2] A timeline or progress bar is displayed.  
[S3] Push notifications alert the customer: “Your order is out for delivery.”

## **UC 6.4 Alternative Flows**

[A1] If the driver's location cannot be updated, the system shows the last known location and a message: "Tracking temporarily unavailable."

---

## **Use Case 7: Customer Contacts Support (Helpdesk)**

### **UC 7.1 Preconditions**

The customer has an active or completed order.

### **UC 7.2 Main Flow**

The customer navigates to the "Help" section. [S1]

The customer selects the order they need help with. [S2]

The system connects them to support (chat, phone, or ticket). [S3] [A1]

### **UC 7.3 Subflows**

[S1] The customer taps "Help" in the app menu.

[S2] They select Order #45 and describe their issue.

[S3] The system either initiates a chat with an agent or confirms a ticket has been created.

### **UC 7.4 Alternative Flows**

[A1] If support agents are unavailable, the system provides self-help FAQs and promises a callback.

---

## **Use Case 8: Restaurant Confirms Order Readiness**

### **UC 8.1 Preconditions**

The restaurant has received an order and is preparing it.

### **UC 8.2 Main Flow**

The kitchen staff finish preparing the meal. [S1]

They mark the order as "Ready for Pickup." [S2]

The system notifies the driver that the order is ready. [S3] [A1]

### **UC 8.3 Subflows**

[S1] Staff complete food prep.

[S2] On the order screen, they tap "Mark Ready."

[S3] The driver's app updates with "Ready for Pickup."

#### **UC 8.4 Alternative Flows**

[A1] If the staff marked “Ready” by mistake, they can revert the order to “In Preparation.”

---

### **Use Case 9: Marketing Team Creates a Basic Promotion**

#### **UC 9.1 Preconditions**

The user is logged in with marketing privileges.

#### **UC 9.2 Main Flow**

The marketer navigates to the promotions section. [S1]

They create a new campaign (e.g., discount code). [S2]

They set campaign details (discount %, validity dates). [S3]

The system activates the promotion. [S4] [A1]

#### **UC 9.3 Subflows**

[S1] The marketer clicks “Promotions.”

[S2] They click “+ New Campaign” and select “Promo Code.”

[S3] They enter “SAVE20, 20% off, valid Sept 1–15.”

[S4] The campaign is published and shown at checkout.

#### **UC 9.4 Alternative Flows**

[A1] If invalid dates or conflicts occur, the system displays: “Campaign dates overlap with an existing promotion.”

---

### **Use Case 10: System Logs Orders for Regulatory Compliance**

#### **UC 10.1 Preconditions**

The system has processed customer orders.

#### **UC 10.2 Main Flow**

The system automatically records essential order data (items, timestamps, payment info). [S1]

The data is stored securely for reporting and auditing. [S2] [A1]

#### **UC 10.3 Subflows**

[S1] At checkout completion, the system logs the order in its compliance database.

[S2] Records include restaurant ID, order items, payment type, and delivery time.

## UC 10.4 Alternative Flows

[A1] If the system cannot log the data (e.g., network outage), it retries periodically until the log is complete.

# Reflection

In order to decide which use cases to keep and which to get rid of, we brainstormed with two LLMs, ChatGPT and Gemini. I will discuss our conversations with both and share how they influenced our thinking.

## Discussion with ChatGPT

When narrowing down options, I often think the easiest first step is to throw out obvious “no’s.” ChatGPT agrees with that sentiment, and advised us to avoid the nice-to-haves that can be added in later. Via this thinking, ChatGPT advised to avoid (for now) use cases that focus on things such as enhancements and convenience (such as multi-language support and advanced analytics).

So now that we’ve cut out some of the more obviously frivolous use cases, we can work from the opposite end and focus on the absolute bare necessities. ChatGPT focused on a minimum viable product being able to form a complete delivery loop. The customer must be able to order, pay, and receive that order. This means that the restaurant must be able to make the order and the delivery driver must be able to deliver it.

We asked ChatGPT to make sure that the most critical stakeholders were included, and so it also provided use cases for support, marketing, and legal.

## Discussion with Gemini

Gemini took this perspective: that an MVP is the bare minimum that will allow the system to function. As someone with a systems background, I quite like this definition. Without wheels, there is no car. Without something providing force to the wheels, there also is no car.

Gemini looked at this challenge from the lens of what it considers to be the four main stakeholder groups: Customer, Restaurant Staff, Operations Staff, and Support Staff. A customer must be able to place an order and make payment. Restaurant staff must be able to see orders, mark them as fulfilled, put items on the menu and take items off the menu (even if temporarily, such as “out of stock”). Delivery drivers must be included in the mix, and customers must have some avenue of customer support.

In deciding which use cases not to include for an MVP, Gemini also focused on excluding the nice-to-haves for now. It also considered feature complexity, and excluded features that would be complicated to implement but not absolutely necessary for the system to function.

One interesting angle that Gemini took that we had not considered: processes that can be manually performed in the background, but that do not need to be in the app at first, can be

excluded from this MVP. For example, a customer needs to be able to select an item to order in the app, but an accountant doesn't necessarily need an app feature to perform their role in the process. As long as the system keeps track of orders and payment, the accountant can complete their tasks offline or on a different platform.

## **Conclusion**

In the end, minimizing use cases to create an MVP is all about the tug and pull between stakeholders. Does the customer need to have an easy and smooth experience? Absolutely. Does there need to be a nice, real-time interface for those completing background, administrative tasks such as finance and compliance? Less so.

Both LLMs agree that the customer, restaurant, and delivery driver are the pieces of the system that are most important for bare functionality. The customer will not submit a paper form to order their sandwich. The accountant can, to start at least, receive an.xls file with order information and still complete their job successfully.

Going off of that, one might think about the different speeds of information flow. Customers need to be able to order instantly, restaurants need to see orders quickly, and drivers need to know where to go right away. These stakeholders have needs that have a high time-priority and so their needs are what we focus on here when narrowing down our use cases.

## **Prompt History**

ChatGPT

<https://chatgpt.com/share/68c81bf4-e3cc-8004-84ff-d7089d9739b9>

Gemini

<https://g.co/gemini/share/0e835e9d13b1>