

## **Use Cases:**

### **1. Register/Login & Manage Profile**

#### **Preconditions:**

- User has internet access and is on the webpage.

#### **Main Flow:**

1. User selects "Sign Up" or "Login."
2. System validates credentials or creates new account.
3. User lands on profile dashboard.

#### **Sub Flow:**

1. User edits personal details (name, address, payment methods).
2. System saves and confirms changes.

#### **Alternate Workflow:**

- Invalid credentials → System prompts retry.
- Network failure → Retry later or store offline.

### **2. Browse Menus & Restaurants**

**Main Case:** User browses restaurants and menus to explore available food options.

#### **Preconditions:**

- User is logged in.
- Restaurants are available in the database.

#### **Main Flow:**

1. User searches or applies filters (cuisine, ratings, location).
2. System loads matching restaurants.

3. User selects a restaurant.

**Sub Flow:**

1. System displays menu with descriptions, prices, and images.
2. User checks ratings and reviews.

**Alternate Workflow:**

- No restaurants available → Show fallback message.
- Menu item unavailable → Notify user with alternatives.

**3. Apply Promo Code/Discounts**

**Main Case:** User applies discount or promo code to an order.

**Preconditions:**

- User has items in the cart.
- Promo code is valid in system.

**Main Flow:**

1. User enters promo code at checkout.
2. System validates promo details.

**Sub Flow:**

1. If valid, system applies discount.
2. Updated total is displayed.

**Alternate Workflow:**

- Invalid/expired code → Show error message.
- Minimum order not met → Notify user.

**4. Track Order Status (GPS)**

**Main Case:** User tracks real-time status of order.

**Preconditions:**

- User has placed an order.
- Driver has been assigned.

**Main Flow:**

1. User navigates to “Order Status.”
2. System shows current stage (“Preparing,” “Picked Up”).

**Sub Flow:**

1. User views courier’s live GPS location and ETA.
2. System sends push notifications.

**Alternate Workflow:**

- GPS unavailable → Show last known location.
- Delivery delayed → Notify with revised ETA.

## **5. Modify Active Order**

**Main Case:** User modifies an active, unconfirmed order.

**Preconditions:**

- User has an active (not yet confirmed) order.

**Main Flow:**

1. User navigates to "Active Orders."
2. User selects "Edit Order."

**Sub Flow:**

1. User adds/removes items.
2. System recalculates total and updates order.

**Alternate Workflow:**

- Order already confirmed → System denies edits and suggests cancellation.

**6. Request Refund/Report Wrong Order**

**Main Case:** User requests refund for wrong or unsatisfactory order.

**Preconditions:**

- User has a completed or canceled order.

**Main Flow:**

1. User submits refund request with details.
2. System logs request and routes to support.

**Sub Flow:**

1. Support reviews and approves refund.
2. Refund is processed to user's account.

**Alternate Workflow:**

- Ineligible refund → System rejects request with reason.
- Support delay → System shows "Pending" status.

**7. Provide Ratings & Reviews**

**Main Case:** User rates and reviews restaurant and delivery experience.

**Preconditions:**

- Order is delivered.

**Main Flow:**

1. System prompts user to leave feedback.
2. User selects rating (1–5 stars) and/or writes a review.

**Sub Flow:**

1. System stores feedback.
2. Review is visible to restaurants/admins.

**Alternate Workflow:**

- Low rating → System triggers support follow-up.
- No rating submitted → Reminder notification.

## **8. Contact Customer Support**

**Main Case:** User contacts support for issues.

**Preconditions:**

- User is logged in.

**Main Flow:**

1. User opens “Help Center.”
2. Selects issue category (delivery, payment, app issue).
3. Mention their Concern in the Textbox.

**Sub Flow:**

1. System provides FAQs or chatbot assistance.
2. If unresolved, user escalates to live agent.

**Alternate Workflow:**

- High wait time → System offers callback or email option.

## **9. Manage Subscriptions (Meal Plans)**

**Main Case:** User manages weekly/monthly food subscription.

**Preconditions:**

- User has subscribed to a plan.

**Main Flow:**

1. User navigates to “Subscriptions.”
2. Views active plan details.

**Sub Flow:**

1. User pauses, cancels, or modifies the plan.
2. System confirms and updates billing.

**Alternate Workflow:**

- Payment issue → System retries or requests update.

## **10. Emergency Order Handling**

**Main Case:** System prioritizes urgent orders (e.g., during crises).

**Preconditions:**

- Surge in orders detected.

**Main Flow:**

1. System identifies high-priority users (e.g., medical staff).
2. Assigns priority delivery.

**Sub Flow:**

1. User notified of priority status.
2. Estimated delivery time updated.

**Alternate Workflow:**

- Too many priority requests → System queues based on urgency.

**11. Driver Login & Availability**

**Main Case:** Driver logs in and sets availability.

**Preconditions:**

- Driver has valid account.

**Main Flow:**

1. Driver opens app and logs in.
2. Marks status as “Available.”

**Sub Flow:**

1. System registers driver as active.
2. Orders can now be assigned.

**Alternate Workflow:**

- Invalid credentials → Retry login.
- No shifts scheduled → Access denied.

**12. Accept/Reject Delivery Requests**

**Main Case:** Driver receives and responds to delivery requests.

**Preconditions:**

- Driver is available.

**Main Flow:**

1. System assigns delivery to nearest driver.
2. Driver receives notification.

**Sub Flow:**

1. Driver accepts request.
2. System updates customer ETA.

**Alternate Workflow:**

- Driver rejects → System reassigns order.
- No response → Timeout leads to reassignment.

**13. Update Delivery Status**

**Main Case:** Driver updates delivery progress.

**Preconditions:**

- Driver has accepted order.

**Main Flow:**

1. Driver updates status at key stages (Picked Up, Out for Delivery, Delivered).
2. System syncs with customer app.

**Sub Flow:**

1. Notifications sent to customer.
2. ETA updated.

**Alternate Workflow:**

- Driver forgets → System auto-updates using GPS logs.

**14. Navigate with Optimized Routes**

**Main Case:** Driver uses system for best delivery route.

**Preconditions:**

- Driver has accepted an order.



**Main Flow:**

1. System calculates optimal route.
2. Driver views navigation in-app.

**Sub Flow:**

1. Driver follows route and updates status.
2. System continuously tracks progress.

**Alternate Workflow:**

- Traffic or closure detected → System recalculates route.

**15. Manage Menu & Pricing**

**Main Case:** Restaurant manages menu offerings and prices.

**Preconditions:**

- Restaurant staff logged into dashboard.

**Main Flow:**

1. Staff navigates to “Menu Management.”
2. Adds, edits, or deletes items.

**Sub Flow:**

1. System syncs updates to customer app.
2. Menu availability updated.

**Alternate Workflow:**

- Invalid input → Prompt correction.
- System downtime → Changes queued.

**16. Update Business Information**

**Main Case:** Restaurant updates business profile.

**Preconditions:**

- Restaurant account verified.

**Main Flow:**

1. Staff opens “Business Info.”
2. Updates details (hours, contact, address).

**Sub Flow:**

1. System validates and saves changes.
2. Updates sync to app.

**Alternate Workflow:**

- Invalid input → Error prompt.
- System failure → Contact admin for manual update.

## **17. Track Payments & Settlements**

**Main Case:** Restaurant tracks financial transactions.

**Preconditions:**

- Completed orders exist.

**Main Flow:**

1. Restaurant accesses “Payments.”
2. Views daily/weekly settlement reports.

**Sub Flow:**

1. System shows successful and pending payments.

2. Staff can export data.

**Alternate Workflow:**

- Settlement delays → Notify restaurant.
- System down → Manual reconciliation required.

**18. Respond to Customer Reviews**

**Main Case:** Restaurant responds to customer ratings and reviews.

**Preconditions:**

- Customers have submitted reviews.

**Main Flow:**

1. Staff opens “Reviews” section.
2. Selects review to respond.

**Sub Flow:**

1. Staff writes reply (public or private).
2. Response saved and displayed.

**Alternate Workflow:**

- Inappropriate review flagged to admin.
- System unavailable → Contact support for manual posting.

**19. Scan & Track Inventory**

**Main Case:** Restaurant scans inventory to update menu availability.

**Preconditions:**

- Restaurant uses barcode or stock management system.

**Main Flow:**

1. Staff scans inventory items.
2. System syncs stock with menu.

**Sub Flow:**

1. Out-of-stock items marked unavailable.
2. Menu updated in real time.

**Alternate Workflow:**

- Scanner/system failure → Manual entry required.

**20. Onboard Home-Based/Mess Provider**

**Main Case:** Individuals start food business on platform.

**Preconditions:**

- User applies for provider account.

**Main Flow:**

1. Applicant submits registration (details, documents).
2. Admin reviews and approves.

**Sub Flow:**

1. Account activated as restaurant.
2. Menu setup enabled.

**Alternate Workflow:**

- Application rejected → Notify applicant with reason.

**21. Monitor System Performance & Analytics**

**Main Case:** Admin monitors platform health and metrics.

**Preconditions:**

- Monitoring tools active.

**Main Flow:**

1. Admin accesses performance dashboard.
2. Views live server/API/database status.

**Sub Flow:**

1. Generates performance reports.
2. Shares insights with stakeholders.

**Alternate Workflow:**

- Dashboard failure → Admin uses backup logs.

**22. Handle Disputes & Refunds**

**Main Case:** Admin resolves customer, driver, or restaurant disputes.

**Preconditions:**

- A dispute is reported.

**Main Flow:**

1. Admin reviews complaint details.
2. Investigates via order logs.

**Sub Flow:**

1. Approves or rejects refund/compensation.
2. Updates stakeholders via system.

**Alternate Workflow:**

- Missing order data → Escalate to backend support.

## **23. Manage Payments & Reconciliations**

**Main Case:** Admin ensures smooth financial settlements.

### **Preconditions:**

- Completed orders exist.
- Payment gateway active.

### **Main Flow:**

1. Admin checks transaction logs.
2. Reviews pending settlements.

### **Sub Flow:**

1. Processes adjustments or refunds.
2. Sends payment confirmation to restaurants.

### **Alternate Workflow:**

- Gateway failure → Retry or switch to backup.
- Payment mismatch → Escalate to finance team.

## **24. Enforce Security & Fraud Detection**

### **Preconditions**

- Payment/data stored in system; security protocols enabled.

### **Main Flow:**

1. System continuously monitors transactions.
2. Anomaly detection flags suspicious activities.
3. Fraudulent transactions blocked; account flagged

### **Subflow**

1. Security module cross-checks with third-party anti-fraud API.

**Alternate Workflow:**

- If false positive → user verification workflow (OTP, ID check).

**25. Support Multi-Role Access (User, Restaurant, Admin)****Preconditions**

- User is authenticated with assigned role.

**Main Flow:**

1. System verifies role (customer, restaurant, admin).
2. Grants access to appropriate dashboard/features.
3. Role-specific operations executed (order food, manage menu, monitor analytics).

**Subflow:**

1. Super-admin assigns or revokes roles dynamically.

**Alternate Workflow:**

- Unauthorized access attempt triggers account lock and alert.

**26. Process Payments via Gateway****Preconditions:**

- User has valid payment method; gateway integrated.

**Main Flow:**

1. Customer places order and proceeds to checkout.
2. System sends request to payment gateway.
3. Gateway processes transaction; success confirmation sent.

**Subflow:**

1. Supports promo code validation before payment confirmation.

**Alternate Workflow:**

- Payment failure → retry or prompt alternate method.
- Gateway timeout → transaction queued and retried.

**27. Integrate Third-Party APIs (Maps, Payment, SMS)****Preconditions:**

- API keys configured; services reachable.

**Main Flow:**

1. App calls third-party service (maps for routing, SMS for OTP, payment gateway).
2. API returns data to system.
3. System processes and presents to user.

**Subflow:**

1. Middleware monitors API performance.

**Alternate Workflow:**

- API failure → system retries, then falls back to backup provider.

**28. Automate Order Notifications (Email/SMS/Push)****Preconditions:**

- Order is placed/updated; user has notification settings enabled.

**Main Flow:**

1. Order event triggered (confirmed, out for delivery, delivered).
2. Notification service pushes email/SMS/app alert.
3. User acknowledges update.



**Subflow:**

1. Notifications aggregated into activity history.

**Alternate Workflow:**

- If notification service down → queued and sent later.

**29. Database Testing, Debugging & Logging****Preconditions:**

- Test environment active; debug server running.

**Main Flow:**

1. Tester executes API/database test cases.
2. System logs responses and validates against expected results.
3. Failures logged with detailed error reports.

**Subflow:**

1. Round-table peer review of debugging outputs.

**Alternate Workflow:**

- If API timeout occurs → retry or flag as failed.

**30. Deploy & Publish Updates****Preconditions:**

- Build tested and approved for release.

**Main Flow:**

1. Deploy tested version to production.
2. Monitor performance post-deployment.
3. Collect feedback and resolve issues.

**Subflow:**

1. Canary release to small user group before full rollout.

**Alternate Workflow:**

- Rollback triggered if severe bugs detected.

**Reflection document (max half page): differences you see in the LLM reports.**

- Lining up the reports, you can tell the models approached the task with different mindsets. ChatGPT's output felt like something I could hand straight to an instructor or an engineer. It called out the roles clearly (admin, staff, customer), worked in the NC 2.0% food tax handling, and covered practical pieces like recipe changes, refunds, pickup confirmations, privacy/consent, and accessibility. The scope was tight and the use cases mapped cleanly to testable criteria.
- Claude's write-up, by contrast, impressed me with how strictly it stuck to the use case structure. Every item came with preconditions, main flow, sub flow, and alternative flows spelled out. It didn't really lag behind ChatGPT in speed, but it stood out for being more disciplined and predictable in presentation. If I needed to scan or compare quickly, Claude's format made that much easier.
- DeepSeek aimed much broader, closer to what you'd expect from a national food-delivery marketplace. It leaned heavily into addresses, driver accept/reject, tracking, reorders, and promo campaigns. That's useful for customer growth scenarios, but it skipped over the governance details—like tax setup and accessibility—that matter more in a café setting. It was slower to output stuff and work with.
- Some models digested the Ollama-generated .pkl context chunks without issue, but Claude didn't. I ended up pulling the data out manually before they could process it from the .pkl file.

**Report the total cost of LLM usage (note: up to \$80 is acceptable, divided 4 ways between 4 team members = \$20 each).**

- We used chatgpt plus: \$20