Project: 1b1

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 Gandamsetty	igandam@ncsu.edu

📄 Use Case: Schedule a Food Order

Description

Customer schedules a food order for a later delivery window.

Preconditions

- Customer is authenticated.
- At least one saved delivery address.

Main Flow

1. Customer builds the cart by selecting menu items and options [Build Cart]
2. Customer selects a future delivery time window [Select Time Window]
3. Customer chooses a payment method and applies promotions if any [Select Payment]
4. Customer reviews the order and places it [Place Order]
5. System confirms the order and sends notifications [Notify Parties]
6. At prep time, the restaurant prepares and marks it ready [Restaurant Prep]
7. A driver is assigned and delivers at scheduled time [Assign and Deliver]
8. Customer tracks the order until completion [Track Order]

Subflows

- [Build Cart] – Customer selects items, customizes modifiers, validates stock, and views subtotal, taxes, fees, and ETA.
- [Select Time Window] – System shows available slots; customer selects one within restaurant hours and cutoff rules.
- [Select Payment] – Customer selects card, wallet, or cash-on-delivery (if allowed); system authorizes payment.
- [Place Order] – System validates cart, slot, address, and serviceability, then creates the order.
- [Notify Parties] – Customer receives confirmation; restaurant gets scheduled order ticket with prep time.
- [Restaurant Prep] – System alerts restaurant at prep lead time; restaurant accepts or flags issues.
- [Assign and Deliver] – Dispatch service assigns driver; driver picks up and delivers at slot.
- [Track Order] – Customer sees status and live tracking; case ends when delivered.

Alternative Flows

- [Slot Unavailable] – Time window becomes unavailable; system prompts customer to choose another slot or restaurant.
- [Item Out of Stock] – An item is unavailable before prep; customer offered substitution/removal with price adjustment.
- [Payment Authorization Failed] – Payment pre-authorization fails; customer retries with another method or order is canceled.
- [Restaurant Cancels] – Restaurant cannot fulfill order; customer notified, refund issued, alternatives suggested.
- [Driver Shortage] – No driver available; system attempts reassignment, proposes new window, or cancels with refund.

📄 Use Case: Contactless Delivery

Description

Driver completes a contactless delivery with proof of drop-off and optional PIN verification.

Preconditions

- Order has been prepared and assigned to a driver.
- Customer has selected contactless delivery and provided drop-off instructions.

Main Flow

1. Driver arrives at the restaurant to pick up the order [Pick Up Order].
2. Driver confirms pickup and starts navigation to the customer [Navigate].
3. Driver follows contactless delivery instructions at the destination [Contactless Drop].
4. Driver captures proof of delivery and confirms completion [Complete Delivery].
5. The system notifies the customer and closes the order [Notify Completion].

Subflows

- [Pick Up Order] The driver verifies the order code/name, checks sealed packaging, and collects required utensils or beverages.
- [Navigate] The driver uses turn-by-turn navigation; the ETA is shared with the customer in real time.
- [Contactless Drop] The driver follows instructions (e.g., “leave at door,” “front desk,” “security gate”), places the package, and rings/knocks if requested.
- [Complete Delivery] The driver takes a time-stamped photo, optionally requests a one-time PIN if required by the merchant or building, and marks the order delivered.
- [Notify Completion] The system updates status, sends photo/proof to the customer, prompts for rating and tip.

Alternative Flows

- [Order Not Ready] The restaurant delays preparation. The driver waits within policy or the order is reassigned after a threshold.
- [Access Restricted] The driver cannot access the building or gated community. The driver contacts the customer for access or redirects to an alternative drop-off point.
- [Customer Unreachable] The driver cannot reach the customer. After multiple attempts and a wait period, the driver follows return or dispose policies.
- [Wrong Address] The GPS location or address is incorrect. The driver confirms the correct location with the customer or support before delivery.
- [Safety Concern] The driver deems the location unsafe. The driver aborts delivery and escalates to support for resolution.

📄 Use Case: Handling Out-of-stock Item.

Description

Restaurant handles an out-of-stock item with a customer-approved substitution.

Preconditions

- Active order contains an item that becomes unavailable during preparation
- Substitute recommendations are configured.

Main Flow

1. The kitchen flags an item as out of stock [Flag OOS].
2. The system generates recommended substitutes [Suggest Subs].
3. The restaurant proposes a substitute to the customer [Propose Sub].
4. The customer approves or rejects the substitution [Customer Decision].
5. The restaurant updates the order accordingly [Update Order].

Subflows

- [Flag OOS] Mark the item unavailable on the order and optionally on the live menu; pause fulfillment until resolved.

- [Suggest Subs] Use menu data and pricing to propose similar items and portion sizes.
- [Propose Sub] Send the top substitute(s) to the customer with any price difference and prep-time impact.
- [Customer Decision] Notify the customer via app push/SMS; collect approval, rejection, or custom request within a time limit. If no response, apply the timeout policy [No Response Policy].
- [No Response Policy] After the timer expires, follow the merchant's configured default (auto-refund item or apply closest equal-priced substitute).
- [Update Order] Apply the selected action, adjust pricing and taxes, notify all parties.

Alternative Flows

- [No Suitable Substitute] Offer to remove the item with a refund and continue the rest of the order.
- [Customer Cancels Order] Cancel the entire order per policy and process a full refund.
- [Price Increase Approval Needed] Require explicit consent for higher-priced substitutes before proceeding.
- [Menu Sync] If multiple items are flagged OOS, trigger a temporary menu pause to prevent new orders until updates complete.

📄 Use Case: Handling Missing Item.

Description

Customer reports a missing item and receives a resolution.

Preconditions

- Order is marked delivered.
- The issue is reported within the support window.
- Payment was captured.

Main Flow

1. The customer opens the delivered order [Open Order].
2. The customer selects “Missing Item” and identifies the item(s) [Select Issue].
3. The system validates eligibility using delivery data and risk checks [Validate].
4. The system presents available resolutions [Offer Resolution].
5. The customer selects a resolution [Choose Resolution].
6. The system applies the resolution and confirms it [Apply Resolution].

Subflows

- [Open Order] Display recent orders and allow selection of the affected delivery.
- [Select Issue] Guide the customer to pick the missing item(s) and add optional photos or notes.
- [Validate] Check delivery photo, bag count, driver notes, and prior claims; score the request for automated handling or review [Route Handling].
- [Route Handling] Direct low-risk cases to instant resolution; send high-risk cases to manual review [Manual Review].
- [Offer Resolution] Show options such as refund to original payment, delivery credit, or redelivery if feasible.
- [Choose Resolution] Capture the customer’s choice and any constraints (availability window for redelivery).
- [Apply Resolution] Execute refund [Process Refund], issue credit [Issue Credit], or create a redelivery order [Create Redelivery].
- [Process Refund] Post a partial refund for the missing item(s) and associated taxes/fees if applicable.
- [Issue Credit] Grant an account credit usable on future orders.
- [Create Redelivery] Place a no-charge replacement order to the original address.

Alternative Flows

- [Outside Window] Inform the customer the claim window has expired and route to support for exceptions.
- [Manual Review] A human agent reviews evidence and decides on the appropriate outcome.
- [Restaurant Contact Needed] Contact the restaurant for confirmation before resolution.
- [Ineligible Item] Exclude items that cannot be replaced (e.g., alcohol) and adjust options.

📄 Use Case: Edits/Cancel Order

Description

Customer edits or cancels an order before preparation starts.

Preconditions

- Order exist and is pending at the restaurant.

Main Flow

1. The customer opens the order details [Open Order].
2. The system checks modifiability [Check Modifiability].
3. The customer applies changes or cancels [Apply Changes].
4. The system confirms updates [Confirm Updates].

Subflows

- [Open Order] Retrieve the order by ID and show current status.
- [Check Modifiability] Determine if the restaurant has not started preparing items.
- [Apply Changes] Add/remove items, change quantity, edit notes, or request cancel.
- [Confirm Updates] Send changes to the restaurant, adjust totals and authorization, notify the customer.

Alternative Flows

- [Already Cooking] The restaurant has started cooking. Disable edits; offer contact support.
- [Price Delta Auth Fail] New total requires higher authorization that fails. Revert changes or change payment.
- [Cancellation Fee] Cancellation triggers a fee per policy. Display fee and ask for confirmation.

📄 Use Case: Payment System

Description

Payment authorization, capture, and fallback handling.

Preconditions

- At least one payment method on file.

Main Flow

1. The system tokenizes the payment method [Tokenize].
2. The system authorizes the order amount [Authorize].
3. The system captures funds on completion [Capture].

Subflows

- [Tokenize] Store a network token via the gateway.
- [Authorize] Request authorization including itemized amounts and tips if applicable.
- [Capture] Capture the final amount after adjustments.

Alternative Flows

- [Auth Failure] Authorization declines. Prompt the customer to retry or change payment.
- [Partial Capture] Adjusted total lower than auth. Capture reduced amount; release remainder.

- [Method Switch] Customer switches payment mid-order. Re-authorize with the new method.

📄 Use Case: Reward System

Description

Apply promo code or loyalty rewards to an order.

Preconditions

- Customer has an active cart.

Main Flow

1. The customer opens available offers [Open Offers].
2. The system checks eligibility [Check Eligibility].
3. The customer applies the chosen offer [Apply Offer].
4. The system recalculates totals [Reprice].

Subflows

- [Open Offers] Display promo codes, vouchers, and loyalty points balance.
- [Check Eligibility] Validate rules: restaurant, basket minimum, time, and user segment.
- [Apply Offer] Attach the benefit to the cart and lock the code for the session.
- [Reprice] Recompute item, tax, fee, and delivery totals and show savings.

Alternative Flows

- [Promo Ineligible] The cart does not meet conditions. Explain requirements and suggest fixes.
- [Promo Expired] Code is expired or exhausted. Recommend active alternatives.
- [Stacking Not Allowed] Multiple offers conflict. Let the customer choose one to keep.

📄 Use Case: Allergen Safe Order

Description

Customer places an allergen-safe order.

Preconditions

- Customer profile exists.
- Restaurant supports allergen handling.

Main Flow

1. The customer sets or confirms allergy preferences [Set Allergy Profile].
2. The customer browses and selects items with allergy filters applied [Allergy-Aware Browse].
3. The restaurant acknowledges the allergen note and readiness to accommodate [Acknowledge Allergen].
4. The order is prepared, packaged with labels, and delivered [Fulfill].

Subflows

- [Set Allergy Profile] The customer records allergens (e.g., peanuts, shellfish) and cross-contact sensitivity; preferences are enforced across sessions.
- [Allergy-Aware Browse] Hide incompatible items, label potential cross-contact, and require confirmation when adding flagged items [Confirm Risk].
- [Confirm Risk] The customer must acknowledge any cross-contact warning to proceed or choose an alternative.
- [Acknowledge Allergen] The restaurant confirms ingredient checks and prep protocols (separate utensils/areas) or declines [Cannot Accommodate].
- [Cannot Accommodate] Inform the customer and suggest alternative restaurants that can accommodate.
- [Fulfill] Attach allergen and ingredient labels, seal packaging, notify the courier of special handling.

Alternative Flows

- [Allergen Info Missing] An item lacks allergen data. Block selection until data is provided or offer a similar item with verified data.
- [Prep Capacity Limit] The kitchen temporarily cannot handle allergen-safe prep. Offer to delay, split the order, or cancel the allergen item.
- [Courier Mix-Up] The courier attempts item substitution. Disallow substitutions for allergen orders and require restaurant re-confirmation.

📄 Use Case: Transparency Report on Nutrition ranking

Description

Platform generates and publishes a transparency report on nutrition-related ranking and promotions.

Preconditions

- The reporting window has closed and data pipelines have completed.

Main Flow

1. The compliance analyst schedules the report run [Schedule Report].
2. The system compiles ranking, promotion, and exposure metrics [Compile Metrics].
3. The analyst reviews disclosures and approves publication [Publish Report].

Subflows

- [Schedule Report] Select the period, platforms (iOS/Android/web), and geography; lock inputs.
- [Compile Metrics] Aggregate the share of healthier items in top positions, promo exposure by health category, and targeting parameters used [Assemble Disclosures].
- [Assemble Disclosures] Generate methodology, criteria references, and change logs for algorithms.

- [Publish Report] Post to the public portal, notify stakeholders, archive datasets.

Alternative Flows

- [Data Gap] Some vendors did not supply nutrition data. Flag coverage rates and exclude from sensitive metrics with footnotes.
- [Algorithm Update] Ranking logic changed mid-period. Split reporting windows and document impacts.
- [Confidentiality Conflict] A metric risks revealing proprietary data. Aggregate to a higher level or suppress with justification.

📄 Use Case: Assign a Delivery Driver

Description:

Assign and dispatch a driver to deliver an order.

Preconditions:

- The order is accepted by the restaurant
- There are active drivers in the service area
- Location services and mapping are available.

Main Flow:

1. The system searches for eligible drivers [Find Drivers].
2. The system sends an offer to a nearby driver [Dispatch Offer].
3. The driver accepts the offer [Accept Offer].
4. The driver navigates to the restaurant [Navigate to Pickup].
5. The driver picks up the order [Pickup Order].
6. The driver navigates to the customer [Navigate to Dropoff].
7. The driver completes the delivery [Deliver Order].

Subflows:

- [Find Drivers] Filter by proximity, vehicle type, capacity, rating, and current load.
- [Dispatch Offer] Send job details (pickup, items, pay, ETA) with a response timer.
- [Accept Offer] On accept within timer, assign the order; otherwise re-offer to another driver [Dispatch Offer].
- [Navigate to Pickup] Open turn-by-turn navigation; update ETA for customer and restaurant.
- [Pickup Order] Verify order ID/QR, check sealed packaging, collect required receipts, and mark “Picked Up.”
- [Navigate to Dropoff] Navigate to the customer address; update customer ETA dynamically.
- [Deliver Order] Follow delivery preference (hand-off or leave-at-door); collect proof (photo/signature/PIN) [Verify Handoff]; the use case ends.
- [Verify Handoff] Capture evidence and confirm completion; trigger payout and notifications.

Alternative Flows:

- [No Driver Available] Expand search radius, increase incentive, or delay dispatch; notify customer with updated ETA and cancellation option.
- [Order Batching] Offer a stack with compatible orders; recalculate route and ETAs before acceptance.
- [Customer Unreachable] Attempt calls/messages; follow SOP (wait, leave-at-door if allowed, or return to restaurant) and document outcome.

📄 Use Case: Refund System

Description:

Customer requests a refund for a missing or incorrect item.

Preconditions:

The order was completed within the refund window.

Main Flow

1. The customer opens the order in history [Open Order].
2. The customer reports an issue [Report Issue].
3. The customer submits details and evidence [Submit Claim].
4. The system reviews and resolves the claim [Resolve Claim].

Subflows:

- [Open Order] Navigate to past orders and select the affected order.
- [Report Issue] Choose issue type (missing item, wrong item, damaged, late).
- [Submit Claim] Specify items, add notes, and upload photos if applicable; then submit for review.
- [Resolve Claim] If eligible and evidence sufficient, auto-approve [Auto Refund]. Otherwise, escalate to support [Escalate].
- [Auto Refund] Apply a refund or credit and notify the customer. The use case ends.
- [Escalate] Route to an agent with order context and evidence. After review, issue resolution and notify the customer [Issue Resolution].
- [Issue Resolution] Approve, partially approve, or deny with rationale and guidance. The use case ends.

Alternative Flows:

- [Photo Required] The claim type requires a photo. Prompt the customer to add evidence before submission.
- [Out-of-Policy] The claim is outside the allowable time or repeats excessively. Inform the customer and provide next steps.
- [Restaurant Disputes] The restaurant disputes the claim. Hold funds, collect additional info, and have support adjudicate.

📄 Use Case: Deposit Return for Returning Reusable Containers

Description:

Customer returns reusable containers for deposit refund during next delivery.

Preconditions:

The restaurant participates in a reusable-container program and the customer has paid a deposit.

Main Flow:

1. The customer opts in to reusable packaging at checkout [Opt In].
2. The system adds a refundable deposit to the order [Add Deposit].
3. On a subsequent order, the customer schedules container pickup [Schedule Pickup].
4. The courier collects containers during delivery [Collect Containers].
5. The system verifies return quantity and condition [Inspect Return].
6. The deposit is refunded in full or in part based on condition [Refund Deposit].
7. The customer’s program status is updated [Update Status].

Subflows:

- [Opt In] The customer chooses “reusable containers” and accepts program terms.
- [Add Deposit] Apply per-container deposit and show it in the cart and receipt.
- [Schedule Pickup] Offer pickup time aligned with next delivery or a standalone pickup.
- [Collect Containers] The courier scans container IDs and confirms count in app.
- [Inspect Return] Automated or manual QC checks for damage or contamination.
- [Refund Deposit] Issue refund to original payment method and notify the customer.

- [Update Status] Adjust the customer’s container balance and program eligibility.

Alternative Flows:

- [No Containers Available] The courier cannot collect; reschedule pickup at no fee once.
- [Containers Damaged] Apply partial or no refund per policy and share photo evidence.
- [Pickup-Only Request] Customer schedules a pickup without a delivery; apply service fee if applicable.

📄 Use Case: Contactless Age Verification For Alcohol

Description:

Customer orders alcohol with contactless age verification.

Preconditions:

The delivery address is in a permitted region, within legal hours, and the merchant is licensed.

Main Flow:

1. The customer adds alcohol to the cart [Add Alcohol].
2. The system initiates digital age verification [Verify Age].
3. The order is marked as “ID-required” with delivery constraints [Mark ID Required].
4. The courier arrives and completes contactless verification [Complete Handoff].
5. The system records proof and closes the order [Record Proof].

Subflows:

- [Add Alcohol] Verify item eligibility and local restrictions before allowing checkout.

- [Verify Age] Capture a government ID and live selfie; run OCR and liveness/face match checks.
- [Mark ID Required] Flag order with “adult signature required,” no-leave-at-door unless verified.
- [Complete Handoff] Customer presents face for contactless match or shows ID through app-safe flow.
- [Record Proof] Store verification token, timestamp, and courier attestation per policy.

Alternative Flows:

- [Verification Failed] Age verification fails; remove alcohol, proceed with non-alcohol items, or cancel.
- [Customer Not Present] No eligible recipient available; return alcohol to merchant and refund per policy.
- [Restricted Window] Legal delivery window ends before arrival; reschedule or cancel alcohol items.

📄 Use Case: AI-Powered Dynamic Menu Pricing

Description

Restaurant optimizes menu item pricing in real-time based on demand patterns, competitor analysis, and inventory levels using machine learning algorithms.

Preconditions

- Historical sales data spanning at least 6 months is available
- Competitor pricing feeds are configured
- Inventory management system is integrated
- AI pricing model has been trained and validated
- Restaurant manager has pricing approval permissions

Main Flow

1. The system analyzes current demand patterns and inventory levels [Analyze Current State]

2. The system fetches competitor pricing data and market trends [Gather Market Intelligence]
3. The AI model generates optimized pricing recommendations [Generate Recommendations]
4. The restaurant manager reviews and approves price changes [Review and Approve]
5. The system updates menu prices across all platforms [Update Pricing]
6. The system monitors performance impact and adjusts [Monitor Performance]

Subflows

- **[Analyze Current State]** System pulls last 30 days of sales data, identifies trending items, calculates demand elasticity, checks current inventory levels, and flags items with unusual patterns [Pattern Detection].
- **[Pattern Detection]** Algorithm identifies seasonal trends, day-of-week patterns, time-of-day demand curves, and correlations between weather/events and sales volume.
- **[Gather Market Intelligence]** System scrapes competitor websites, processes third-party pricing feeds, analyzes local market conditions, and incorporates economic indicators [Market Analysis].
- **[Market Analysis]** Calculate competitor price positioning, identify pricing gaps, assess market saturation, and factor in local purchasing power data.
- **[Generate Recommendations]** AI model processes all inputs, applies demand forecasting algorithms, calculates optimal price points for profit maximization, and generates confidence scores for each recommendation [Price Optimization].
- **[Price Optimization]** Algorithm considers profit margins, inventory turnover goals, customer price sensitivity, and competitive positioning to suggest price adjustments between -20% to +15% of current prices.
- **[Review and Approve]** System presents recommendations in dashboard with projected impact, allows manager to modify suggestions, requires approval for changes exceeding preset thresholds [Manager Approval], and logs all decisions.

- **[Manager Approval]** Display price change rationale, show projected revenue impact, highlight risk factors, and require explicit confirmation for implementation.
- **[Update Pricing]** System pushes approved prices to POS system, updates online ordering platforms, synchronizes with delivery apps [Platform Sync], and notifies staff of changes.
- **[Platform Sync]** Coordinate price updates across website, mobile app, third-party delivery platforms, and in-store displays to ensure consistency.
- **[Monitor Performance]** Track sales volume changes, monitor customer feedback, measure revenue impact, and feed results back into AI model for continuous learning [Performance Analytics].
- **[Performance Analytics]** Calculate price elasticity in real-time, measure customer retention impact, assess competitor responses, and identify successful pricing strategies.

Alternative Flows

- **[Data Quality Issues]** Historical data incomplete or corrupted. Use statistical interpolation and reduce recommendation confidence.
- **[Competitor Data Unavailable]** Pricing feeds fail. Rely on historical data and increase weight of internal demand patterns.
- **[Inventory System Offline]** Cannot access stock levels. Use last known inventory and apply conservative pricing adjustments.
- **[Model Accuracy Degraded]** AI performance drops. Reduce recommendation aggressiveness and flag model for retraining.
- **[Manager Override]** Manager consistently rejects recommendations. Analyze patterns and adjust model parameters.
- **[Platform Sync Failure]** Price updates fail on platforms. Retry updates and alert manager to discrepancies.
- **[Customer Complaints Spike]** Negative feedback increases. Reduce pricing aggressiveness and suggest promotional campaigns.
- **[Peak Demand Override]** High-demand periods trigger surge pricing with customer notification and automatic post-event reversion.

📄 Use Case: Customer Reports Order Issue and Requests Resolution

Description:

A customer reports problems with their delivered order and initiates a resolution process through the customer service system.

Preconditions:

- Customer has a completed order in the system
- Order was delivered within the last 24 hours
- Customer is authenticated in their account
- Customer service module is operational

Main Flow:

1. Customer accesses order history and selects problematic order [Access Order History]
2. Customer selects issue type and provides detailed description [Report Issue]
3. Customer uploads supporting evidence if available [Submit Evidence]
4. System creates support ticket and estimates resolution time [Create Ticket]
5. Customer service representative reviews case and contacts relevant parties [Review Case]
6. System processes resolution and applies compensation if approved [Process Resolution]
7. Customer receives notification of resolution and provides feedback [Close Case]

Subflows:

- [Access Order History] Customer navigates to past orders and identifies the problematic delivery
- [Report Issue] Customer selects from categories like wrong items, missing items, quality issues, or late delivery

- [Submit Evidence] Customer can attach photos of incorrect/damaged items or provide additional context
- [Create Ticket] System assigns unique ticket ID, categorizes issue severity, and sends confirmation to customer
- [Review Case] Representative validates complaint, contacts restaurant/driver if needed, and determines appropriate resolution
- [Process Resolution] System applies refund, credit, or reorder based on resolution decision and updates customer account
- [Close Case] Customer rates resolution experience and case is archived with outcome data

Alternative Flows:

- [Duplicate Report] If customer already reported the same issue, system merges tickets and notifies customer
- [Restaurant Dispute] If restaurant contests the complaint, case is escalated to senior support with additional investigation
- [Fraudulent Claim] If system detects suspicious reporting patterns, case is flagged for fraud review before processing
- [Immediate Escalation] For severe issues like food safety concerns, case is immediately escalated to management with priority handling

📄 Use Case: Staff Manages Real-Time Inventory Updates

Description:

Restaurant staff updates item availability and inventory levels in real-time to prevent overselling and maintain accurate menu displays for customers.

Preconditions:

- Staff member is authenticated with inventory management permissions
- Restaurant is actively receiving orders through the system
- Menu items have established inventory tracking parameters
- System connectivity is stable

Main Flow:

1. Staff accesses inventory management dashboard during service hours
2. Staff reviews current stock levels and pending order queue [Check Current Status]
3. Staff identifies items running low or out of stock [Inventory Assessment]
4. Staff updates item availability and quantities in the system [Update Inventory]
5. System automatically adjusts menu displays and notifies affected customers [Propagate Changes]

Subflows:

- [Check Current Status] Staff views real-time inventory levels, preparation times, and order volume metrics
- [Inventory Assessment] Staff evaluates remaining portions, ingredient availability, and kitchen capacity constraints
- [Update Inventory] Staff marks items as unavailable, reduces quantities, or adjusts preparation times
- [Propagate Changes] System removes unavailable items from customer view and updates estimated delivery times

Alternative Flows:

- [Emergency Stock Depletion] If popular item suddenly runs out during peak hours, system offers automatic substitution suggestions to pending orders.
- [Supplier Delivery Received] If new inventory arrives mid-service, staff can quickly restore item availability and notify waitlisted customers.
- [Kitchen Equipment Failure] If equipment breaks down affecting specific menu categories, staff can bulk-disable related items and activate maintenance mode notifications.

📄 Use Case: System Generates Dynamic Delivery Route Optimization

Description:

The system automatically calculates and optimizes delivery routes for multiple orders to minimize delivery times and maximize driver efficiency during peak service periods.

Preconditions:

- Multiple confirmed orders are ready for delivery assignment
- Delivery drivers are logged in and available for assignments
- GPS and mapping services are operational
- Traffic and weather data feeds are accessible

Main Flow:

1. System collects all pending delivery orders within scheduling window [Gather Order Data]
2. System analyzes delivery addresses, order priorities, and driver locations [Route Analysis]
3. System calculates optimal multi-stop routes considering constraints [Generate Routes]
4. System assigns routes to available drivers and sends notifications [Dispatch Orders]
5. System monitors delivery progress and adjusts routes for remaining orders [Track and Adapt]

Subflows:

- [Gather Order Data] System compiles order details, delivery addresses, time windows, and special instructions
- [Route Analysis] System processes geographic data, traffic patterns, driver capacity, and delivery time commitments
- [Generate Routes] Algorithm creates efficient multi-delivery routes while respecting promised delivery windows
- [Dispatch Orders] System sends route details to driver mobile app with turn-by-turn navigation and customer contact info

Alternative Flows:

- [Driver Unavailable] If assigned driver becomes unavailable, system automatically reassigns orders to next optimal driver
- [Traffic Disruption] If major traffic incident occurs, system recalculates affected routes and notifies customers of delays
- [Order Modification] If customer changes delivery address after route assignment, system evaluates impact and may reassign to different driver for efficiency

📄 Use Case: Admin Analyzes Delivery Performance Metrics

Description:

System administrator reviews delivery performance data to identify bottlenecks, optimize routes, and improve customer satisfaction ratings.

Preconditions:

- Admin has system analytics access permissions.
- Historical delivery data exists for analysis period.
- Delivery driver location tracking is enabled.
- Customer feedback data is available.

Main Flow:

1. Admin selects reporting timeframe and performance metrics [Select Analysis Parameters]
2. System generates delivery performance dashboard with key indicators [Generate Performance Report]
3. Admin identifies underperforming routes or time periods [Analyze Bottlenecks]
4. Admin creates optimization recommendations and policy updates [Develop Improvements]
5. System schedules automated reports and alerts for ongoing monitoring [Setup Monitoring]

Subflows:

- [Select Analysis Parameters] Admin chooses date range, geographic zones, driver teams, and specific KPIs to analyze
- [Generate Performance Report] System calculates average delivery times, customer ratings, order accuracy, and driver efficiency metrics
- [Analyze Bottlenecks] Admin drills down into problem areas using heat maps, trend analysis, and comparative data
- [Develop Improvements] Admin documents findings and creates action items for route optimization or staffing adjustments
- [Setup Monitoring] System configures real-time alerts for performance thresholds and automated weekly summary reports

Alternative Flows:

- [Insufficient Data] If selected timeframe has insufficient delivery data, system suggests alternative date ranges or warns about statistical significance
- [System Performance Impact] If complex analytics queries slow system response, admin receives option to schedule report generation during off-peak hours
- [Export Failure] If admin attempts to export large datasets and process fails, system offers data segmentation options or alternative formats

📄 Use Case: Customer Schedules Recurring Weekly Order

Description:

A customer sets up an automated weekly order for regular meal delivery to streamline their routine food ordering process.

Preconditions:

- Customer has an active account with verified payment method
- Customer has previously placed at least one successful order
- Selected restaurant supports recurring order functionality
- System scheduling service is operational

Main Flow:

1. Customer accesses their order history and selects "Create Recurring Order"
2. Customer chooses items and customizes recurring order template [Build Template]
3. Customer sets delivery schedule and frequency preferences [Configure Schedule]
4. Customer reviews recurring order summary and confirms setup [Confirm Recurring Setup]
5. System creates recurring order profile and schedules first delivery [Initialize Recurring Order]

Subflows:

- [Build Template] Customer selects items from previous orders or creates new combination, sets default quantities and customizations
- [Configure Schedule] Customer chooses delivery day, time window, frequency (weekly/bi-weekly), and duration of recurring service
- [Confirm Recurring Setup] Customer reviews total cost, delivery schedule, and modification/cancellation policies
- [Initialize Recurring Order] System creates automated job, sends confirmation, and schedules first order processing

Alternative Flows:

- [Menu Item Discontinued] If recurring order item becomes unavailable, system notifies customer and requests substitution or order modification
- [Payment Method Expired] If stored payment method expires, system pauses recurring orders and prompts customer to update payment information
- [Delivery Address Changed] If customer moves, system allows address update and verifies delivery availability for new location

📄 Use Case: Admin Implements Dynamic Surge Pricing

Description:

System administrator configures and activates surge pricing during peak demand periods to balance order volume and delivery capacity.

Preconditions:

- Admin has surge pricing management permissions
- Historical order and delivery data is available in the system
- Restaurant partners have agreed to surge pricing participation
- Real-time demand monitoring system is active

Main Flow:

1. Admin analyzes current demand patterns and delivery capacity metrics [Assess Demand Conditions]
2. Admin configures surge pricing rules and thresholds [Set Surge Parameters]
3. Admin activates surge pricing for affected areas and time periods [Enable Surge Pricing]
4. System applies dynamic pricing adjustments and notifies customers [Apply Price Changes]
5. Admin monitors surge pricing effectiveness and adjusts as needed [Monitor Performance]

Subflows:

- [Assess Demand Conditions] Admin reviews real-time order volume, driver availability, average delivery times, and customer wait times
- [Set Surge Parameters] Admin defines price multipliers, geographic zones, duration limits, and automatic deactivation triggers
- [Enable Surge Pricing] System validates configuration, sends notifications to participating restaurants, and updates customer-facing pricing
- [Apply Price Changes] System displays surge pricing notifications to customers, calculates adjusted totals, and processes orders at surge rates
- [Monitor Performance] Admin tracks order conversion rates, customer complaints, driver utilization, and revenue impact

Alternative Flows:

- [Customer Backlash] If customer complaints exceed threshold, admin can immediately disable surge pricing and issue service credits
- [Driver Shortage Persists] If surge pricing doesn't improve delivery capacity, system escalates to emergency protocols and partner driver recruitment
- [Technical Malfunction] If surge pricing algorithm malfunctions and applies incorrect rates, system automatically reverts to standard pricing and logs incident

📄 Use Case: Admin Monitors System Performance and Generates Analytics Report

Description:

System administrator monitors platform health, analyzes business metrics, and generates comprehensive reports for stakeholders and decision-making.

Preconditions:

- Admin has elevated system access privileges
- System monitoring tools are operational
- Historical data is available for analysis
- Reporting infrastructure is functioning

Main Flow:

1. Admin accesses administrative dashboard with system metrics [Access Admin Panel]
2. Admin reviews real-time performance indicators and alerts [Monitor Performance]
3. Admin configures and generates custom analytics reports [Generate Reports]
4. Admin analyzes trends and identifies optimization opportunities [Analyze Data]

5. Admin distributes reports to stakeholders and archives results [Distribute Results]

Subflows:

- [Access Admin Panel] Authenticate with multi-factor authentication and load comprehensive system overview
- [Monitor Performance] Check server response times, transaction success rates, user activity, and error logs
- [Generate Reports] Select date ranges, metrics, filters, and output formats for custom business intelligence reports
- [Analyze Data] Review customer behavior patterns, restaurant performance, peak usage times, and revenue trends
- [Distribute Results] Email automated reports to stakeholders, export data for external analysis, and maintain audit trails

Alternative Flows:

- [Performance Degradation] If system metrics indicate issues, admin can trigger automated scaling, alert technical teams, or initiate maintenance mode
- [Data Anomaly Detected] If unusual patterns are identified, admin can investigate potential fraud, system bugs, or data integrity issues
- [Compliance Audit] If regulatory reporting is required, admin can generate specialized compliance reports with required documentation and validation

📄 Use Case: Staff Handles Kitchen Equipment Failure During Service

Description:

Restaurant staff manages kitchen equipment breakdown by disabling affected menu items, activating maintenance mode notifications, and coordinating with repair services while maintaining customer service.

Preconditions:

- Staff member has kitchen management permissions

- Equipment monitoring system is operational
- Menu items are properly categorized by required equipment
- Customer orders may be in progress

Main Flow:

1. Staff identifies equipment failure through alerts or physical inspection [Equipment Assessment]
2. Staff determines which menu categories are affected by the breakdown [Impact Analysis]
3. Staff bulk-disables all menu items requiring the failed equipment [Disable Menu Items]
4. Staff activates maintenance mode notifications for customers [Customer Communication]
5. Staff coordinates repair service and estimates restoration timeline [Repair Coordination]

Subflows:

- [Equipment Assessment] Staff verifies equipment status, documents failure type, and checks safety protocols
- [Impact Analysis] System identifies all menu items dependent on failed equipment and calculates revenue impact
- [Disable Menu Items] Staff selects equipment category and system automatically disables related items across all ordering channels
- [Customer Communication] System displays maintenance notifications on app/website and sends alerts to customers with pending orders
- [Repair Coordination] Staff contacts approved repair vendors, schedules service, and updates estimated repair completion time

Alternative Flows:

- [Partial Equipment Function] If equipment has limited functionality, staff can disable only specific menu items while keeping others available
- [Alternative Preparation Method] If backup equipment exists, staff can reassign affected items to alternative cooking methods and update

preparation times

- [Emergency Closure] If critical equipment fails affecting food safety, staff can temporarily close restaurant and notify all customers with active orders

📄 Use Case: Food Employee Reports Illness and Requests Work Restriction

Description:

A food service employee reports symptoms of potential foodborne illness and follows proper protocols for work restriction and medical clearance.

Preconditions:

- Employee is experiencing symptoms consistent with foodborne illness
- Employee has access to company illness reporting procedures
- Person in Charge (PIC) is available to receive illness report
- Local health officer contact information is accessible

Main Flow:

1. Employee recognizes symptoms and contacts Person in Charge immediately [Symptom Recognition]
2. PIC conducts illness assessment using standardized questionnaire [Illness Evaluation]
3. PIC determines appropriate work restrictions based on symptoms [Restriction Determination]
4. Employee provides detailed exposure and symptom timeline [Exposure Documentation]
5. PIC notifies local health officer of potential foodborne illness case [Health Authority Notification]
6. Employee seeks medical evaluation and diagnostic testing [Medical Consultation]
7. Employee provides medical clearance before returning to work [Return Authorization]

Subflows:

- [Symptom Recognition] Employee identifies fever, diarrhea, vomiting, or other symptoms listed in company policy
- [Illness Evaluation] PIC asks about symptom onset, severity, recent food consumption, and potential exposure sources
- [Restriction Determination] PIC restricts employee from food handling based on specific symptoms and exposure risk
- [Exposure Documentation] Employee provides information about recent meals, travel, and contact with ill persons
- [Health Authority Notification] PIC contacts local health department within required timeframe and provides case details
- [Medical Consultation] Employee visits healthcare provider for diagnosis and appropriate treatment
- [Return Authorization] Employee obtains written medical clearance and negative test results before resuming food handling duties

Alternative Flows:

- [Confirmed Outbreak Exposure] If employee was exposed to confirmed foodborne illness outbreak, extended restriction and additional testing required
- [Hepatitis A Exposure] If employee has hepatitis A exposure, immediate exclusion from work and vaccination requirements apply
- [Asymptomatic Shedding] If employee tests positive but is asymptomatic, work restrictions continue until consecutive negative tests obtained
- [Emergency Staffing] If illness creates critical staffing shortage, PIC implements contingency plans and may request temporary staff
- [Workers Compensation] If illness is determined to be work-related, employee files workers compensation claim and follows occupational health protocols

📄 Use Case: Health Inspector Conducts Food Safety Compliance Audit

Description:

A health inspector performs a comprehensive food safety audit of a restaurant to ensure compliance with FDA Food Code requirements and Active Managerial Control (AMC) practices.

Preconditions:

- Inspector has valid credentials and authority to conduct audit
- Restaurant is operational during business hours
- Person in Charge (PIC) is present at the establishment
- Audit checklist and documentation tools are available

Main Flow:

1. Inspector arrives and identifies themselves to the Person in Charge [Initial Contact]
2. Inspector reviews food safety documentation and certifications [Document Review]
3. Inspector observes food handling practices and employee hygiene [Operational Observation]
4. Inspector checks food storage temperatures and equipment calibration [Temperature Verification]
5. Inspector evaluates Active Managerial Control implementation [AMC Assessment]
6. Inspector documents findings and discusses violations with PIC [Violation Documentation]
7. Inspector provides compliance timeline and follow-up requirements [Compliance Planning]

Subflows:

- [Initial Contact] Inspector presents credentials, explains audit scope, and requests to speak with certified Person in Charge
- [Document Review] Inspector examines employee health records, temperature logs, supplier certifications, and training documentation

- [Operational Observation] Inspector watches food preparation, checks handwashing compliance, and verifies proper use of gloves and utensils
- [Temperature Verification] Inspector uses calibrated thermometer to check refrigeration units, hot holding equipment, and food internal temperatures
- [AMC Assessment] Inspector evaluates preventative food safety measures, staff training effectiveness, and hazard monitoring systems
- [Violation Documentation] Inspector records critical and non-critical violations with photographic evidence and detailed descriptions
- [Compliance Planning] Inspector sets correction deadlines, schedules re-inspection if needed, and explains appeal process

Alternative Flows:

- [Critical Violation Found] If critical food safety violations are discovered, inspector may require immediate cessation of affected operations
- [PIC Unavailable] If certified Person in Charge is not present, inspector may postpone audit or conduct limited inspection with available staff
- [Equipment Malfunction] If temperature monitoring equipment fails during inspection, inspector uses backup calibrated instruments
- [Language Barrier] If communication issues arise, inspector arranges for translator or bilingual staff member to assist
- [Establishment Closure] If imminent health hazards are identified, inspector initiates emergency closure procedures and notifies management

📄 Use Case: Customer Tracks Order and Provides Delivery Feedback

Description:

A customer monitors their order progress in real-time and provides feedback about delivery experience to improve service quality.

Preconditions:

- Customer has placed a confirmed order with valid order ID
- Order is assigned to delivery driver and in progress

- Customer has access to tracking interface via app or web
- GPS tracking and communication systems are functional

Main Flow:

1. Customer accesses order tracking interface using order confirmation [Access Order Tracking]
2. Customer views real-time order status and estimated delivery time [View Order Progress]
3. Customer receives notifications about order milestones and updates [Receive Status Updates]
4. Customer communicates with driver if needed for delivery instructions [Contact Driver]
5. Customer confirms order delivery and reviews items received [Confirm Delivery]
6. Customer provides rating and feedback about delivery experience [Submit Feedback]
7. System processes feedback and updates service quality metrics [Process Feedback]

Subflows:

- [Access Order Tracking] Customer enters order number or logs in to view active orders with live tracking map
- [View Order Progress] System displays order preparation status, driver location, and updated delivery estimates
- [Receive Status Updates] Customer gets push notifications for order confirmed, being prepared, picked up, and out for delivery
- [Contact Driver] Customer uses in-app messaging or calling feature to provide specific delivery instructions or address clarifications
- [Confirm Delivery] Customer verifies all items received match order and marks delivery as complete
- [Submit Feedback] Customer rates delivery speed, food quality, driver service, and provides optional written comments

- [Process Feedback] System aggregates ratings, flags issues for restaurant/driver attention, and updates recommendation algorithms

Alternative Flows:

- [Delivery Delay] If delivery exceeds estimated time significantly, customer receives compensation offer and updated timeline
- [Missing Items] If items are missing from delivery, customer reports issue and receives refund or replacement order
- [Delivery Address Issues] If driver cannot locate address, customer receives call/message to provide directions or meet at alternate location
- [Order Damaged] If food arrives damaged or incorrect, customer can report issue with photos and receive immediate resolution

---

📄 Use Case: Running payout cycles

Description

Admin settles commissions and payouts for the period.

Preconditions

- The payout window has ended.
- Finance admin is authenticated.

Main Flow

1. The admin reviews all completed transactions for the period [Review Transactions].
2. The admin reconciles gateway settlements and chargebacks [Reconcile].
3. The admin applies commission rules and adjustments [Calculate Commissions].
4. The admin generates statements for restaurant and drivers [Generate Statements].
5. The admin triggers bulk payouts via the payment provider [Trigger Payouts].

6. The admin notifies stakeholders with statements and payout confirmations [Notify Stakeholders].

Subflows

- [Review Transactions] Load orders, fees, tips, promos, refunds, and taxes for the window.
- [Reconcile] Match platform records to gateway batches and bank reports; flag mismatches for investigation [Handle Exceptions].
- [Calculate Commissions] Apply tiered commissions, promotions, and local taxes to derive net amounts payable/receivable.
- [Generate Statements] Produce itemized statements and invoices, archive for audit, and expose them in portals.
- [Trigger Payouts] Initiate ACH/wallet payouts in batches and monitor statuses until completion.
- [Notify Stakeholders] Send statements and payout confirmations to restaurants, drivers, and finance.
- [Handle Exceptions] Hold affected lines, create support/risk tickets, and resolve before re-run.

Alternative Flows

- [Gateway Outage] The provider is down. Queue payouts for retry and inform stakeholders of delays.
- [Negative Balance] A partner’s balance is negative due to refunds/chargebacks. Net against future payouts or issue an invoice.
- [Fraud Suspicion] Anomalies detected (e.g., wash orders). Freeze accounts, hold payouts, and escalate to risk for investigation.

📄 Use Case: Enabling Healthier Preferences.

Description

User enables a healthier choices preferences to influence recommendations and defaults.

Preconditions

- User Account exists.
- App supports nutrition tags for some items.

Main Flow

1. The user opens account settings [Open Preferences].
2. The user enables the healthier choices preference [Toggle Healthy Mode].
3. The user defines health criteria [Define Criteria].
4. The system updates the recommendation and ranking algorithm [Update Recommendations].
5. The system saves the settings and applies them [Save and Apply].

Subflows

- [Open Preferences] The user navigates to Settings > Food & Health and loads current preferences.
- [Toggle Healthy Mode] The user turns on Healthy Mode. The app explains that healthier restaurants/items will be prioritized and defaults adjusted.
- [Define Criteria] The user selects options (e.g., max calories per item, vegetarian/vegan, lower sodium, whole grains, no sugar-sweetened beverages) and can limit exposure to unhealthy promotions.
- [Update Recommendations] The system re-ranks restaurants and items to prioritize those meeting the criteria, sets default sides/drinks in bundles to healthier options, and limits add-on prompts to healthier items.
- [Save and Apply] Preferences are stored; a filter chip is shown on search pages; during checkout, suggested swaps appear for items exceeding criteria.

Alternative Flows

- [Missing Nutrition Data] If an item lacks nutrition info, the app displays a notice and either excludes it from prioritization or asks the user to include it anyway.
- [Conflicting Criteria] Selected criteria conflict (e.g., keto and vegan). The app prompts the user to relax one constraint.

- [Child Account Detected] For teen/child profiles, the app restricts unhealthy promotions and applies stricter defaults per policy.
- [Opt-Out] The user turns off Healthy Mode. The app restores standard ranking and optionally requests feedback.

📄 Use Case: Remit Transaction Taxes

Description

Remit transaction taxes, including local meals taxes, for the filling period.

Preconditions

- Jurisdiction mappings, registrations, calendars, and payment accounts have been initialized.

Main Flow

1. At period end, the system closes the tax period and locks transactions [Close Period].
2. The system aggregates taxable bases and taxes by jurisdiction [Aggregate Transactions].
3. The system validates nexus and filing obligations [Validate Obligations].
4. The system prepares state and local returns and payment instructions [Prepare Returns].
5. The system submits returns and remits payments [Remit Payments].
6. The system archives filings and schedules next-period tasks [Archive & Schedule].

Subflows

- [Close Period] Freeze edits to tax-relevant fields, version the dataset, and record exchange rates if needed.
- [Aggregate Transactions] Roll up item and fee taxes by state and locality; separate special meals/meals-and-rooms liabilities from general sales tax.
- [Validate Obligations] Reassess thresholds/elections; if thresholds newly exceeded, create registrations and backfill accruals where required.

- [Prepare Returns] Populate state forms, county meals tax forms where required, and attach supporting schedules (fee taxability, exemptions).
- [Remit Payments] Execute ACH/credit card payments to appropriate state and, where mandated, county authorities; capture confirmations.
- [Archive & Schedule] Store copies of returns, receipts, and audit trails; queue next filing reminders and rate updates.

Alternative Flows

- [Jurisdiction Mismatch] A transaction maps to an unknown locality code. Route to exception handling, correct mapping, and recalculate. [Aggregate Transactions].
- [Payment Failure] A remittance fails. Retry per policy; if unresolved, notify finance and the authority, and initiate contingency payment.
- [Audit Notice] An authority issues an audit or desk review. Freeze related records and export requested data with reconciliations.

📄 Use Case: Refilling consumer-owned containers at pickup

Description

Restaurant seeks approval to refill consumer-owned containers.

Preconditions

- The regulatory authority accepts written contamination-free plans.

Main Flow

1. The manager drafts a refill plan [Draft Plan].
2. The manager submits the plan to the regulator [Submit Plan].
3. The regulator reviews and decides [Approval Decision].
4. The restaurant implements the approved process [Implement Process].

Subflows

- [Draft Plan] Define eligible foods and containers, contamination controls, and cleaning steps [Define Controls]; identify equipment needs [Assess Equipment]; and staff procedures [Train Staff].
- [Define Controls] Show how cross-contamination is prevented for consumer- and establishment-refilled containers.
- [Assess Equipment] Identify gravity bins, fill stations, and sanitizing tools needed.
- [Train Staff] Create SOPs on inspection of consumer containers, refilling steps, and incident handling.
- [Submit Plan] Provide the written plan and required documentation to the regulatory authority.
- [Approval Decision] If the plan meets requirements, approval is granted [Approve]; otherwise, revisions are requested [Revise].
- [Approve] Receive written approval with any conditions.
- [Revise] Update the plan per regulator feedback and resubmit.
- [Implement Process] Acquire equipment, train staff, and begin service; maintain records and monitor.

Alternative Flows

- [Plan Rejected] If the plan is denied, suspend the initiative and escalate for technical assistance.
- [Equipment Delayed] Use a phased rollout or postpone launch until equipment arrives.
- [Consumer Misuse] If a consumer container is visibly soiled or unsuitable, refuse refill and offer a sanitary alternative.

📄 Use Case: Bare-Hand Contact Program

Description

Restaurant implements a bare-hand contact program with active managerial control.

Preconditions

- The establishment has a Certified Food Protection Manager and a designated Person in Charge.

Main Flow

1. The CFPM assesses eligibility for bare-hand contact [Assess Eligibility].
2. The CFPM documents policies and logs [Document Controls].
3. The establishment requests approval if required [Request Approval].
4. The PIC trains staff and monitors compliance [Train and Monitor].

Subflows

- [Assess Eligibility] Determine which foods qualify (e.g., RTE ingredients that will be cooked to required temperatures) and which require barriers.
- [Document Controls] Write SOPs for handwashing, no-bare-hand exceptions, and illness reporting; create illness and monitoring logs [Create Logs].
- [Create Logs] Establish 90-day retention for illness reporting logs when bare-hand contact with uncooked RTE foods is approved.
- [Request Approval] Submit documentation to the regulator when approval is required and await decision.
- [Train and Monitor] Train employees, observe practices, and perform corrective actions [Corrective Action] as needed.
- [Corrective Action] Reinforce training, discard contaminated food, and document steps taken.

Alternative Flows

- [Illness Reported] Exclude or restrict ill employees per policy; update logs and adjust staffing.
- [Noncompliance Found] Voluntarily discontinue bare-hand contact, retrain staff, and request reinstatement approval if required.
- [Audit Failure] If regulatory inspection finds deficiencies, implement a corrective action plan and verify effectiveness.

📄 Use Case: Sustainability-Driven Delivery Model

Description

Founder pilots “GreenWave” low-emission batched delivery windows to differentiate on sustainability and reduce cost.

Preconditions

- The platform has baseline unit economics and emissions per order measured.

Main Flow

1. The founder defines the hypothesis and success metrics [Define Hypothesis].
2. Operations models the unit economics impact of batched routes and time-window delivery [Model Unit Economics].
3. Partnerships recruits a pilot restaurant cohort and secures packaging commitments [Recruit Cohort].
4. Product enables green-slot UI and carbon labels; Legal reviews claims [Build Product].
5. Investors review capital needs for the pilot and agree tranche gates [Investor Review].
6. The platform runs an A/B test in two districts for 6 weeks [Run Experiment].
7. Leadership decides whether to scale, iterate, or sunset based on KPIs [Decision].

Subflows

- [Define Hypothesis] Specify target conversion lift, CAC change, AOV impact, NPS delta, and emissions reduction per order [Experiment Metrics].
- [Model Unit Economics] Build sensitivity on drop density, courier incentives, SLA variance, and customer churn risk.
- [Recruit Cohort] Sign MOUs with restaurants including sustainable packaging standards [Sustainable Packaging] and co-marketing fund contributions [Co-Marketing Fund].

- [Build Product] Add selectable green windows, carbon labels tied to route density, and in-cart nudges; validate marketing claims with counsel [Claims Compliance].
- [Investor Review] Share pro forma runway and payback; align on capital release milestones tied to [Experiment Metrics].
- [Run Experiment] Randomize eligible customers; monitor conversion, on-time rate, courier earnings stability, carbon per order.
- [Decision] If thresholds are met, expand to additional zones with a phased courier incentive plan; otherwise iterate pricing or messaging, or sunset.

Alternative Flows

- [Greenwashing Risk] Regulators challenge carbon claims. Remove labels, keep green windows, and engage a third-party verifier before re-enabling [Claims Compliance].
- [Courier Pushback] Couriers resist batching due to earnings variability. Introduce guaranteed-floor incentives and opt-in windows; if still negative, reduce batch aggressiveness.
- [Restaurant Non-Compliance] A partner misses packaging standards. Remove green badge and shift volume to compliant venues; offer remediation training.

📄 Use Case: Nutrition Summary

Description

Restaurant publishes compliant nutrition disclosures and healthfulness scores aligned with government-endorsed criteria.

Preconditions

- Government nutrient profiling criteria are configured on the platform.
- Restaurant staff and nutritionists have access.
- Labeling placements are standardized.

Main Flow

1. Nutritionist staff enter recipes, ingredients, and portions [Enter Menu Data].
2. Public health officer monitors compliance dashboards [Monitor Compliance].
3. The platform publishes labels and summaries consistently across listings [Publish Labels].

Subflows

- [Enter Menu Data] Provide ingredients, prep methods, portion sizes, allergens, and kid’s meal defaults.
- [Validate Nutrition] Run analysis for calories, sodium, sugars, saturated fat; confirm allergens; recommend adjustments [Reformulate].
- [Score Items] Apply the approved profiling scheme (e.g., traffic light/health stars); lock “healthy” descriptors to items meeting criteria.
- [Monitor Compliance] Track missing labels, claim misuse, and high-sodium flags; issue corrective actions and timelines [Notify Vendor].
- [Publish Labels] Display summary nutrition next to item name/price in equal prominence; link to full panels and ingredients.

Alternative Flows

- [Missing Data] Block item publishing or promotions until labels are complete; notify restaurant.
- [High Sodium Flag] Suggest compliant sides, smaller portions, or recipe changes [Reformulate].
- [Child-Facing Marketing] Prevent unhealthy items from appearing in child-targeted placements or sponsored units.

📄 Use Case: Dual Plan Exit Strategy

Description

Board runs a dual-track exit (strategic sale vs. IPO/SPAC) to maximize valuation and strategic fit.

Preconditions

- The company has reached scale with audited financials
- Investment banks and legal counsel are engaged.

Main Flow

1. The board defines success criteria and constraints [Set Objectives].
2. The company selects advisors and forms a deal team [Select Advisors].
3. The team prepares materials: data room, S-1 draft, and buyer deck [Prepare Materials].
4. Advisors run a confidential market check with strategics and PE sponsors [Market Check].
5. Finance builds valuation models and adjusts for key risks [Valuation Work].
6. The board compares offers vs. public comps and chooses a path [Choose Path].
7. The company negotiates definitive agreements [Negotiate Definitives].
8. The transaction closes; post-close integration or public readiness continues [Close and Transition].

Subflows

- [Set Objectives] Align on valuation range, consideration mix, employee outcomes, brand independence, and regulatory risk tolerance.
- [Select Advisors] Engage banks with both M&A and ECM strength; appoint counsel experienced in antitrust and labor matters.
- [Prepare Materials] Populate cohort LTV/CAC, city-level contribution, regulatory exposure, and ESG metrics; prepare working capital analyses.
- [Market Check] Solicit IOIs, explore JV/majority/100% sale, and SPAC/PIPE interest; preserve confidentiality.
- [Valuation Work] Run DCF, scenario analysis (labor regulation, price war), and sum-of-the-parts (core delivery, advertising, logistics SaaS).
- [Negotiate Definitives] Structure earn-outs tied to unit economics, address founder lock-ups, and secure regulatory approvals [Regulatory Review].

Alternative Flows

- [Antitrust Scrutiny] A strategic buyer triggers a second request. Extend outside date, consider divestiture remedies, or pivot to IPO.
- [Market Window Shuts] Equity markets deteriorate. Pause IPO leg and pursue a structured growth round with secondary liquidity.
- [PIPE Falls Through] SPAC financing gaps emerge. Downsize deal, add earn-out, or shift to private sale.

Reflection Document: differences you see in the LLMs report

Claude API

The **Claude API** demonstrated strong reliability and precision when working with the provided documents.

- **Fact-Adherence:**

  When asked about topics not covered in the source material (e.g., *"car insurance use cases"*), Claude correctly responded that it had no information rather than fabricating an answer. This highlights its commitment to factual accuracy, which is valuable in professional and compliance-focused contexts.

- **Structured Output:**

  When given a **detailed prompt with a specific use case template**, Claude followed the instructions exactly. It generated a **new use case for the food delivery system** that adhered to the required format, including key sections such as:

  - Preconditions
  - Main Flow
  - Alternative Flows

- **Consistency and Predictability:**

  Claude's ability to consistently follow complex instructions and produce predictable, structured outputs makes it particularly well-suited for tasks where precision and format compliance are critical.

ChatGPT API

ChatGPT API excelled in areas where **creativity, breadth of coverage, and iterative refinement** were needed. While Claude was more conservative, ChatGPT was better at **exploring novel directions** when carefully guided.

- **Breadth and Exploration:**

  With **careful prompt crafting** informed by document clusters, ChatGPT produced **unique and wide-ranging use cases** that went beyond generic food ordering and delivery. It generated scenarios in areas such as regulatory compliance, sustainability initiatives, financial settlements, and technical system reliability — dimensions that Claude did not naturally explore as deeply.

- **Iterative Flexibility:**

  ChatGPT was highly responsive to **iterative refinement** of prompts. Starting from zero-shot prompts, results were repetitive (mostly reusing ordering/delivery flows). However, once scoped prompts were crafted (e.g., focusing on *business strategy, technical architecture, or regulatory frameworks*), ChatGPT adapted quickly and produced **richer, domain-specific use cases**. This flexibility shows its strength for projects where exploration and refinement are part of the workflow.

- **Creative but Controlled Outputs:**

  While Claude excelled at strict adherence to a given template, ChatGPT balanced **creativity with structure**. With explicit instructions, it could stick to the format (Description, Preconditions, Main Flow, Subflows, Alternative Flows), while still contributing fresh, innovative angles.

Summary of Differences

- **Claude:** Reliable, cautious, strictly factual, and precise — ideal when predictable, controlled outputs are needed.
- **ChatGPT:** Creative, broad, and adaptive — ideal for generating **novel, wide-ranging, and stakeholder-specific use cases** when guided with well-crafted prompts.

Both APIs have distinct strengths: Claude is better at being conservative and fact-bound, while ChatGPT shines in **exploration, iteration, and creative expansion of ideas**.

Total Cost of LLM Usage

- **Claude API**: $5 (credit purchase)
- **ChatGPT API**: $5 (basic plan subscription)

Total spent by team: $10

Divided among 4 members: $2.50 each