P1C1 - Use Cases

S2G1 - Authors: Robert Kemp, Matthew Nguyen, Tiehang Zheng, Rishi Jeswani

UC-14: Cancel Order

Actors: Customer, Restaurant, Driver, Admin
Description: Customer cancels an order before or during preparation/delivery with appropriate refund handling.

Preconditions:

- Customer has placed an order
- Customer is logged into the platform
- Order is in a cancellable state

Main Flow:

1. Customer opens order tracking or order history
2. Customer selects "Cancel Order" option
3. System checks order status and cancellation policy
4. System calculates refund amount based on timing and progress
5. Customer confirms cancellation
6. System processes refund and notifies all parties
7. Order status updated to "Cancelled"

Subflows:

- **Policy Check:** System applies cancellation rules based on order stage
- **Partial Refund:** System calculates partial charges for preparation costs
- **Notification Chain:** Restaurant and driver (if assigned) receive cancellation alerts

Alternate Flows:

- A1: Order already in preparation → system offers partial refund only
- A2: Driver already dispatched → cancellation blocked, customer contacted
- A3: Restaurant already started cooking → customer pays preparation fee

Postconditions:

- Order cancelled and refund processed
- All parties notified of cancellation

UC-16: Handle Customer Support Requests

Actors: Customer, Support Agent, Admin
Description: Customers submit support tickets for order issues, account problems, or general inquiries.

Preconditions:

- Customer has access to support system
- Support agents available

Main Flow:

1. Customer accesses help/support section
2. Customer selects issue category (order, payment, account, technical)
3. Customer describes problem and submits ticket
4. System assigns ticket to appropriate support agent
5. Support agent reviews and responds to customer
6. Issue resolved and ticket closed

Subflows:

- **FAQ Integration:** System suggests relevant FAQ articles before ticket creation
- **Live Chat:** Customer can engage in real-time chat with support
- **Escalation:** Complex issues escalated to senior agents or admin

Alternate Flows:

- A1: No agents available → system queues ticket with ETA
- A2: Customer unsatisfied with resolution → ticket escalated
- A3: Duplicate ticket detected → system merges with existing

Postconditions:

- Customer issue documented and resolved
- Support metrics updated

UC-17: Apply and Manage Promotional Codes

Actors: Customer, Admin, Marketing Team
Description: Customers apply discount codes during checkout while admins manage promotional campaigns.

Preconditions:

- Valid promotional codes exist in system
- Customer has items in cart

Main Flow:

1. Customer enters promo code during checkout
2. System validates code against current promotions
3. System applies discount and updates order total
4. Customer completes order with discounted price
5. System tracks promo code usage

Subflows:

- **Code Validation:** System checks expiry, usage limits, and eligibility
- **Stacking Rules:** System applies multiple codes according to policy
- **Usage Tracking:** System monitors redemption rates and effectiveness

Alternate Flows:

- A1: Invalid/expired code → system shows error and suggests alternatives
- A2: Usage limit reached → system informs customer code is no longer valid
- A3: Minimum order not met → system explains requirements

Postconditions:

- Valid discounts applied to order
- Promotional usage tracked for analytics

UC-19: Manage Favorites and Saved Items

Actors: Customer
Description: Customers save favorite restaurants and menu items for quick reordering.

Preconditions:

- Customer logged into account
- Customer has browsed restaurants/items

Main Flow:

1. Customer browses restaurants or menu items
2. Customer selects "Add to Favorites" for preferred items

3. System saves favorites to customer profile
4. Customer can access favorites from dedicated section
5. Customer can reorder directly from favorites list

Subflows:

- **Quick Reorder:** Customer adds entire previous order to cart
- **Favorites Organization:** Customer creates lists (work lunch, weekend treats)
- **Availability Check:** System shows if favorite items are currently available

Alternate Flows:

- A1: Favorite restaurant closed → system suggests similar alternatives
- A2: Favorite item discontinued → system recommends replacements
- A3: Storage limit reached → customer must remove items to add new ones

Postconditions:

- Customer preferences saved and easily accessible
- Enhanced user experience for repeat orders

UC-20: Send Order Notifications

Actors: System, Customer, Driver, Restaurant
Description: System sends timely notifications about order status changes to relevant parties.

Preconditions:

- Users have notification preferences set
- Valid contact information available

Main Flow:

1. Order status changes in system
2. System identifies notification recipients based on status
3. System formats appropriate message for each recipient type
4. System sends notifications via preferred channels (push, SMS, email)
5. System logs notification delivery status

Subflows:

- **Multi-Channel Delivery:** System sends via multiple channels for critical updates
- **Customization:** Users can set preferences for notification types and timing
- **Retry Logic:** Failed notifications automatically retried

Alternate Flows:

- A1: Notification delivery fails → system tries alternative contact method
- A2: User opted out of notifications → critical updates still sent
- A3: System overload → notifications queued and sent in batches

Postconditions:

- All relevant parties informed of order status
- Communication preferences respected

UC-21: Handle Out-of-Stock Items

Actors: Restaurant, Customer, System
Description: System manages situations when ordered items become unavailable during order processing.

Preconditions:

- Customer has placed order with specific items
- Restaurant discovers item unavailability

Main Flow:

1. Restaurant marks item as out-of-stock during order preparation
2. System identifies affected active orders
3. System notifies customers of unavailable items
4. System offers alternatives: substitution, removal, or order cancellation
5. Customer makes selection
6. System updates order and processes any refund adjustments

Subflows:

- **Automatic Substitution:** System suggests similar items based on customer preferences
- **Partial Fulfillment:** Customer chooses to proceed with remaining items
- **Full Cancellation:** Customer cancels entire order due to key item unavailability

Alternate Flows:

- A1: Customer doesn't respond to substitution offer → system applies default policy
- A2: Multiple items out-of-stock → system offers order cancellation first
- A3: Substitute item costs more → customer approves additional charge

Postconditions:

- Order adjusted to reflect available items
- Customer satisfaction maintained through proactive communication

---

UC-22: Manage Delivery Time Slots

Actors: Customer, Restaurant, System
Description: Customers schedule deliveries for specific time slots based on restaurant and delivery capacity.

Preconditions:

- Restaurant supports scheduled delivery
- Customer placing order

Main Flow:

1. Customer selects "Schedule for Later" during checkout
2. System displays available time slots based on restaurant capacity
3. Customer selects preferred delivery time
4. System reserves slot and adjusts restaurant preparation schedule
5. Order queued for preparation at appropriate time
6. Normal delivery process begins at scheduled time

Subflows:

- **Capacity Management:** System limits concurrent orders per time slot
- **Preparation Timing:** System calculates when to start cooking for on-time delivery
- **Slot Modification:** Customer can change time slot before preparation begins

Alternate Flows:

- A1: Preferred slot full → system suggests nearby available times
- A2: Restaurant closes before scheduled time → system alerts customer to reschedule
- A3: Customer wants to move to immediate delivery → system checks availability

Postconditions:

- Delivery scheduled for customer's preferred time
- Restaurant workflow optimized for scheduled orders

UC-24: Track Driver Performance

Actors: Admin, Driver
Description: System monitors and evaluates driver performance metrics for quality assurance.

Preconditions:

- Driver has completed deliveries
- Performance data available

Main Flow:

1. System continuously collects driver performance data
2. Admin accesses driver performance dashboard
3. System displays metrics: delivery time, customer ratings, completion rate
4. Admin reviews performance trends and identifies issues
5. Admin takes action for underperforming drivers (coaching, warnings, deactivation)

Subflows:

- **Automated Alerts:** System flags drivers with concerning metrics
- **Performance Trends:** System shows improvement or decline patterns
- **Comparative Analysis:** System ranks drivers for recognition programs

Alternate Flows:

- A1: Driver contests performance rating → admin reviews individual cases
- A2: System error in tracking → admin manually adjusts metrics
- A3: Driver shows improvement → admin removes warnings/restrictions

Postconditions:

- Driver performance documented and managed
- Service quality maintained across platform

UC-25: Manage Group Orders

Actors: Order Organizer (Customer), Group Members (Customers), Restaurant
Description: Multiple customers collaborate on a single order from the same restaurant.

Preconditions:

- Order organizer has account

- Restaurant supports group orders
- Group members have platform access

Main Flow:

1. Order organizer creates group order and selects restaurant
2. System generates shareable link for group order
3. Organizer shares link with group members
4. Group members add items to shared cart
5. Organizer reviews final order and handles payment
6. Single order placed with restaurant for group delivery

Subflows:

- **Individual Payments:** Each member pays for their own items
- **Split Payment:** Group divides total cost equally or by custom amounts
- **Order Coordination:** System prevents conflicts and manages item limits

Alternate Flows:

- A1: Member adds item after organizer starts checkout → system allows last-minute additions
- A2: Payment fails from one member → organizer can cover or remove items
- A3: Restaurant minimum not met → system suggests additional items

Postconditions:

- Group order successfully placed and paid
- Single delivery coordinated for entire group

UC-26: Handle Delivery Address Validation

Actors: Customer, System, Driver
Description: System validates delivery addresses for accuracy and serviceability before order confirmation.

Preconditions:

- Customer entering delivery address
- Address validation service available

Main Flow:

1. Customer enters or selects delivery address

2. System validates address format and existence
3. System checks address against delivery coverage zones
4. System estimates delivery time and fees for validated address
5. Customer confirms address and proceeds with order

Subflows:

- **Auto-Complete:** System suggests addresses as customer types
- **GPS Integration:** Customer's current location auto-fills address fields
- **Delivery Zone Check:** System confirms restaurant delivers to specified area

Alternate Flows:

- A1: Address not found → system requests clarification or manual verification
- A2: Address outside delivery zone → system suggests pickup or nearby restaurants
- A3: Ambiguous address → system presents multiple options for customer selection

Postconditions:

- Valid, serviceable delivery address confirmed
- Accurate delivery estimates provided to customer