PROJECT 1a1

1. **Create list of stakeholders**

   Primary Stakeholders

   These stakeholders are essential for the core food delivery transaction to occur. Without any one of these, the service cannot function.

   - **Customers / Users:** The individuals who initiate the entire process by placing an order and providing the revenue.
   - **Restaurants / Food Providers:** The businesses that supply and prepare the food, fulfilling the product side of the transaction.
   - **Delivery Drivers / Couriers:** The individuals who execute the logistics, physically connecting the restaurant and the customer to complete the service.

   Secondary Stakeholders

   These stakeholders have a significant interest or influence on the platform but are not directly part of the core order-and-delivery transaction itself. They enable, support, fund, or regulate the primary function.

   - **Investors / Shareholders:** Provide the financial capital required to build and scale the business, with an interest in its profitability and growth.
   - **Customer Support Team:** A supporting function that assists primary stakeholders when issues arise during a transaction but is not involved in a successful order.
   - **Marketing & Sales Team:** A business function responsible for growth by acquiring new customers and restaurant partners, influencing the scale of the operation rather than executing it.
   - **Regulatory Authorities:** Government bodies that set the legal and safety rules (e.g., food safety, labor laws) within which the platform must operate.
   - **Key Technology Partners:** External providers of essential services like payment gateways and GPS/mapping services. While critical, they are third-party enablers of the transaction, not core participants.

2. Identify stakeholder biases

Stakeholder A	Stakeholder B	Conflict/ Irrelevance	Description
Customers	Restaurants	Pricing vs Quality	Customers want affordability and speed; restaurants may resist discounts or rushed prep.
Delivery Drivers	Restaurants	Timing Misalignment	Drivers expect ready orders; restaurants prioritize kitchen flow, causing pickup delays.
Delivery Drivers	Customers	Efficiency vs Preference	Drivers optimize routes; customers may expect personalized delivery (e.g., contactless).
Restaurants	Marketing Team	Expectation Mismatch	Marketing may overpromise menu items or delivery speed, misaligning with restaurant reality.
Investors	Regulatory Authorities	Growth vs Compliance	Investors push for scale and cost-efficiency; regulators enforce labor and safety standards.
Customer Support	Delivery Drivers	Advocacy Tension	Support may favor customers in disputes, potentially undermining driver autonomy.
Technology Partners	Delivery Drivers	System Reliability	GPS/payment failures affect drivers; tech partners may not prioritize driver experience.
Restaurants	Technology Partners	Integration Complexity	POS/menu syncing issues can disrupt restaurant operations and order accuracy.
Investors	Delivery Drivers	Strategic Irrelevance	Drivers focus on daily logistics; investor concerns are long-term and financial.

Regulatory Authorities	Marketing Team	Operational Irrelevance	Marketing rarely intersects with regulation unless misrepresentation occurs.
Customer Support	Investors	Functional Irrelevance	Support handles micro-level issues; investors focus on macro-level performance.
Customers	Technology Partners	Interface Irrelevance	Customers rarely interact with backend tech unless there's a visible failure.
Restaurants	Customer Support	Transactional Irrelevance	In smooth operations, restaurants have no need for support team interaction.
Delivery Drivers	Marketing Team	Role Irrelevance	Drivers handle logistics; marketing focuses on acquisition and branding.

3. Comment on prompt crafting

Zero-shot prompting is quick and easy, but the downside is that the responses often feel generic, incomplete, or too basic, meaning you usually need to refine them afterward. Since the prompt doesn't guide the model much, it tends to fall back on broad knowledge and produce answers that aren't always tailored to your exact needs. Careful prompting, on the other hand, is more deliberate. You spend extra time structuring the request, adding context, and being specific about format, tone, or the type of detail you want. That upfront effort pays off because the model delivers responses that are richer, more relevant, and already aligned with your goals, so you spend less time editing. In practice, I've noticed that when I just use zero-shot prompts, the model gives me something serviceable but generic. But when I take the time to carefully design the prompt, the output feels customized, almost like it was written exactly for me. The difference isn't just in detail, but also in how much the answer matches the style, depth, and structure I'm aiming for.

4. Write at least 10 use cases (≈5 pages total):

UC1: Customer Registration and Account Management

Actor: Customer

Preconditions:

- Customer has access to the WolfCafe app or website
- Valid email address and phone number available

Main Flow:

1. Customer selects "Create Account" option
2. System displays registration form
3. Customer enters personal information (name, email, phone, address)
4. Customer creates secure password
5. System sends verification email/SMS
6. Customer verifies account through provided link/code
7. System activates account and displays welcome message
8. Customer can update profile information and preferences

Alternative Flows:

- [E1] If email already exists, system prompts to login or reset password
- [E2] If verification fails, system resends verification code
- [E3] If required fields missing, system highlights incomplete information

UC2: Browse Menu and View Item Details

Actor: Customer

Preconditions:

- Menu items are available and properly configured
- Customer has access to the app

Main Flow:

1. Customer opens WolfCafe menu section
2. System displays categorized menu items with images and prices
3. Customer can filter items by category, dietary restrictions, or price range
4. Customer selects specific item for detailed view
5. System displays comprehensive item information (ingredients, nutrition, allergens)
6. Customer can view customer reviews and ratings
7. Customer can customize item options if available

Alternative Flows:

- [E1] If items are temporarily unavailable, system shows "out of stock" status
- [E2] If menu is being updated, system displays loading state

UC3: Add Items to Cart and Manage Orders

Actor: Customer

Preconditions:

- Customer is logged into their account
- Menu items are available for ordering

Main Flow:

1. Customer selects desired menu item
2. Customer chooses quantity and customization options
3. Customer adds item to cart
4. System updates cart total and item count
5. Customer can modify quantities or remove items from cart
6. Customer can add special instructions for preparation
7. System displays running total with taxes and fees
8. Customer can save cart for later or proceed to checkout

Alternative Flows:

- [E1] If item becomes unavailable after adding to cart, system notifies and offers alternatives
- [E2] If cart exceeds maximum order value, system displays warning

UC4: Process Payment and Complete Order

Actor: Customer

Preconditions:

- Customer has items in cart
- Payment method is available
- Restaurant is accepting orders

Main Flow:

1. Customer reviews order summary and pricing breakdown
2. Customer selects pickup time or delivery option
3. Customer chooses payment method (card, digital wallet, etc.)
4. Customer applies any available coupons or loyalty discounts

5. Customer selects tip amount (15%, 20%, 25%, or custom)
6. System processes payment securely
7. System generates order confirmation number
8. Customer receives confirmation via email/SMS with pickup details
9. System notifies restaurant of new order

Alternative Flows:

- [E1] If payment fails, system retries and offers alternative payment methods
- [E2] If restaurant is closed/busy, system suggests alternative pickup times

UC5: Track Order Status and Receive Notifications

Actor: Customer

Preconditions:

- Customer has placed a confirmed order
- Order is in restaurant's preparation queue

Main Flow:

1. System displays order status dashboard to customer
2. Customer can view estimated preparation time
3. System sends status updates as order progresses (received, preparing, ready)
4. Customer receives push notification when order is ready
5. System provides pickup instructions and location details
6. Customer can contact restaurant directly if needed

Alternative Flows:

- [E1] If order is delayed, system updates estimated time and notifies customer
- [E2] If order cannot be fulfilled, system processes refund and notifies customer

UC6: Manage Menu Items and Inventory

Actor: Restaurant Staff

Preconditions:

- Staff member is authenticated with appropriate permissions
- Restaurant management system is accessible

Main Flow:

1. Staff accesses menu management dashboard
2. Staff can create new menu items with descriptions, prices, and images
3. Staff sets ingredient requirements and preparation instructions
4. Staff updates inventory levels for ingredients and tracks usage
5. Staff can mark items as available, unavailable, or seasonal
6. Staff sets daily specials and promotional pricing
7. System updates customer-facing menu in real-time

Alternative Flows:

- [E1] If ingredient inventory is low, system suggests marking related items unavailable
- [E2] If item changes affect pending orders, system alerts staff to potential issues

UC7: View and Process Customer Orders

Actor: Restaurant Staff

Preconditions:

- Customer orders exist in the system
- Staff has order management access

Main Flow:

1. Staff views incoming orders dashboard sorted by order time
2. Staff can see order details, special instructions, and customer preferences
3. Staff confirms order and begins food preparation
4. Staff updates order status as preparation progresses
5. Staff marks order as "Ready for Pickup" when completed
6. System automatically notifies customer of order completion
7. Staff can add preparation notes or communicate delays

Alternative Flows:

- [E1] If ingredients are insufficient, staff can contact customer for substitutions
- [E2] If preparation time exceeds estimate, staff updates customer with new timeline

UC8: Handle Customer Pickup and Order Completion

Actor: Restaurant Staff and Customer

Preconditions:

- Order has been prepared and marked ready
- Customer arrives within pickup window

Main Flow:

1. Customer arrives at restaurant pickup area
2. Customer provides order number or name for verification
3. Staff locates and verifies order details against customer information
4. Staff hands over completed order to customer
5. Customer confirms order completeness and accuracy
6. Staff marks order as "Picked Up" in system
7. System completes transaction and may prompt customer for feedback

Alternative Flows:

- [E1] If customer doesn't arrive within window, staff follows late pickup policy
- [E2] If order is incorrect, staff prepares replacement or processes partial refund

UC9: Generate Reports and Analytics

Actor: Restaurant Management

Preconditions:

- Management has administrative access
- Historical order and sales data exists

Main Flow:

1. Management accesses reporting dashboard
2. Management selects date ranges and report types
3. System generates sales reports, popular items analysis, and customer trends
4. Management can view peak hours, revenue patterns, and inventory usage
5. System provides insights on customer behavior and order patterns
6. Management can export data for external analysis
7. Management uses insights for menu optimization and staffing decisions

Alternative Flows:

- [E1] If insufficient data exists for selected period, system suggests alternative ranges
- [E2] If report generation fails, system queues request and notifies when complete

UC10: Manage Customer Support and Issue Resolution

Actor: Restaurant Management/Staff

Preconditions:

- Customer has submitted feedback or complaint
- Staff has access to customer service tools

Main Flow:

1. Staff receives customer issue notification through system
2. Staff reviews order history and customer complaint details
3. Staff investigates issue and determines appropriate resolution
4. Staff communicates with customer via app messaging or phone
5. Staff processes refund, replacement order, or other compensation
6. Staff documents resolution in customer service system
7. System tracks issue resolution for quality improvement
8. Staff follows up to ensure customer satisfaction

Alternative Flows:

- [E1] If issue requires management approval, system escalates automatically
- [E2] If customer is unsatisfied with resolution, system provides additional escalation options

UC11: Student Meal Plan Integration

Actor: Student

Preconditions:

- Student is enrolled with active meal plan
- Student has available meal plan credits/swipes
- Meal plan system integration is active

Main Flow:

1. Student logs in using university credentials

2. System displays current meal plan balance and available swipes
3. Student browses menu with items marked as "meal plan eligible"
4. Student adds items to cart and sees real-time meal plan deduction preview
5. Student confirms order using meal plan credits
6. System automatically deducts from university meal plan account
7. Student receives confirmation showing remaining balance

Alternative Flows:

- [E1] If insufficient meal plan balance, system offers partial payment options
- [E2] If meal plan system offline, system queues transaction for later processing
- [E3] If item not eligible for meal plan, system suggests alternatives

UC12: Flexible Student Payment Options

Actor: Student

Preconditions:

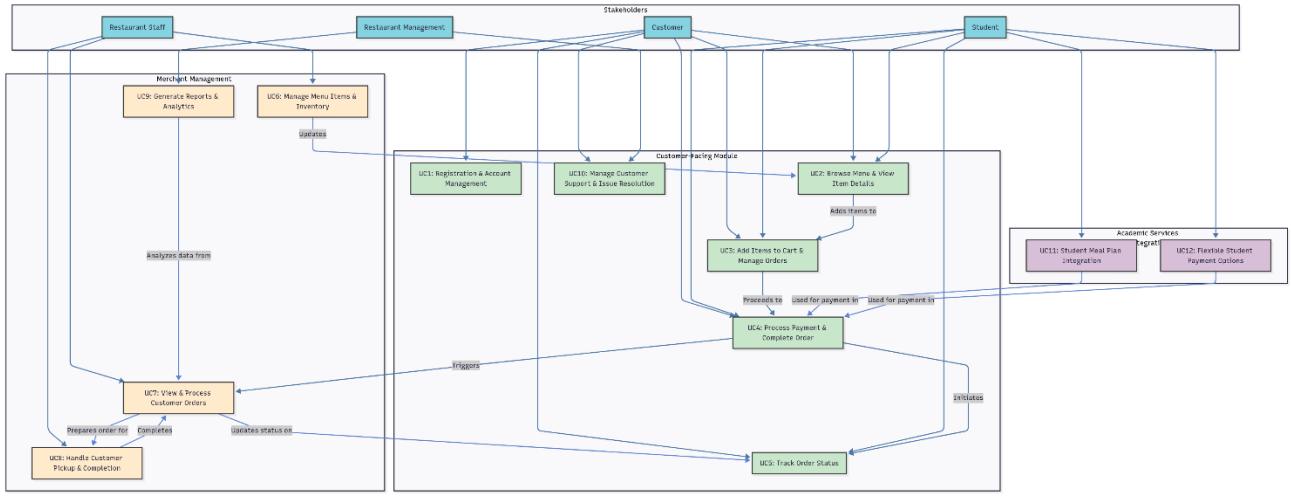
- Student has multiple payment sources available
- Each payment method has sufficient balance for partial payment

Main Flow:

1. Student adds items to cart and proceeds to payment
2. System displays all available payment options with current balances
3. Student allocates payment across methods (50% meal plan, 30% student account, 20% card)
4. System validates each payment source has sufficient funds for allocated amount
5. Student confirms split payment allocation
6. System processes payments sequentially and provides combined receipt
7. Student can save payment preferences for future orders

Alternative Flows:

- [E1] If one payment method fails, system redistributes to remaining methods
- [E2] If total exceeds all available balances, system suggests removing items
- [E3] If preferred allocation invalid, system suggests optimal distribution


Stakeholders
Restaurant Staff
Restaurant Management
Customer
Student
Merchant Management
UC9: Generate Reports & Analytics
UC6: Manage Menu Items & Inventory
Updates
Customer-Facing Module
UC1: Registration & Account Management
UC10: Manage Customer Support & Issue Resolution
UC2: Browse Menu & View Item Details
Adds Items to
UC3: Add Items to Cart & Manage Orders
Analyzes data from
Academic Services
UC11: Student Meal Plan Integration
UC12: Flexible Student Payment Options
Proceeds to
Used for payment in
Used for payment in
UC4: Process Payment & Complete Order
Triggers
UC7: View & Process Customer Orders
Initiates
Prepares order for
Completes
Updates status on
UC8: Handle Customer Pickup & Completion
UC5: Track Order Status