SE PROJECT 1A1 - PROBLEM FAMILIARIZATION

CSC 510 Group 27

Samarth Shah, Harsha Puvvadi, Jai Ramani, Pratham Patel

Product Stakeholders:

1. **Customer** - browses restaurants/menu, places order, receives updates.
2. **Staff** - Manages menu items, fulfills orders, updates prep time.
3. **Admin** - Manages users & restaurants, sets fees, charges & taxes, oversees the system.
4. **Marketing** - Promotes product, attracts users, advertises offers.
5. **Wolfcafe Owner** - Expects revenue growth & profit, Streamlines operational efficiency, Monitors reports & performances.
6. **UI/UX Designer** - Develops User Interface, improves user experience, makes it visually appealing.
7. **Developers** - Build the product, add features, maintain software, improve reliability & performance.
8. **Testers/QA** - Test features & functionality, validate requirements, build testcases & report bugs.
9. **Food Service Specialist** - Ensures good practices are being followed.
10. **Accessibility Expert** - Ensures compliance with WCAG guidelines.
11. **Regulators/Legal** - Safety, fairness, compliance.
12. **Delivery riders** - pick up orders from the restaurant, deliver food.

Stakeholder biases/conflicts:

Conflict 1: Code Quality vs Speed to Market

1. **Developer (Code Quality) vs. Cafe Owner (Speed to Market):**
   A **Developer** wants to write clean, well-documented code using modern technologies. This ensures the system is stable and easy to maintain in the future. The **Cafe Owner**, however, is focused on launching the system as quickly and cheaply as possible to start generating revenue.

2. **Customer (Privacy) vs. Marketing Team (Data):**
   A **Customer** wants a simple, private transaction. They want to order food with minimal data sharing and no intrusive marketing. The **Marketing Team**, on the other hand, needs to collect data on ordering habits, contact information, and user behavior to create targeted promotions and increase sales.

3. **Accessibility Expert (Compliance) vs. UI/UX Designers (Aesthetics):**
   A **UI/UX Designer** wants to create a beautiful, brand-aligned interface using subtle color palettes and minimalist styles. The **Accessibility Expert**, on the other hand, needs to

ensure the application is usable by everyone, which requires strong color contrast for text and highly visible focus indicators for navigation.

4. **Overloaded Staff (Burdened) vs. Owner (Maximizing Profit):**
   An **Overloaded Staff** member, burdened by high-volume work, needs a system that is fast, reliable, and reduces their cognitive load to prevent errors and burnout. The **Owner**, on the other hand, is focused on maximizing profit, which can mean minimizing immediate costs like hiring extra staff for peak hours or investing in faster hardware.

5. **Delivery Driver (Efficiency) vs. Kitchen Staff (Sequence):**
   A **Delivery Driver** needs the system to group orders based on delivery location to create the most efficient routes, saving time and fuel. The **Kitchen Staff**, however, needs to fulfill orders in the chronological sequence they are received to ensure fairness and a "first-in, first-out" workflow.

Prompt Crafting Outcomes:

Zero-Shot Prompting: Direct task requests without specific instructions or examples, relying on model's inherent knowledge base for task completion.

Chain-of-Thought (Structured) Prompting: Step-by-step guided prompts providing explicit reasoning frameworks and contextual examples to enhance output quality and consistency.

Comparative Performance Analysis

Model	Zero-Shot Performance	Structured (Chain-of-Thought) Performance
Gemini	Limited scope adherence; summarized requirements without expansion; unable to analyze external links effectively	Highly effective; expanded scope significantly; reorganized stakeholder classifications with detailed insights
Perplexity (GPT-5)	Strong identification capabilities; provided relevant external links; limited by proprietary resource access	Excellent context maintenance; integrated web-sourced insights; occasional over-extension of requirements
Claude	Well-structured outputs; identified biases and use cases beyond initial scope; integrated external source analysis	Enhanced external source integration; clear organizational structure; reduced creative output for specialized conflicts

Use Cases

UC-01: Customer Places Order with Multiple Items
Primary Actor: Customer
Stakeholders & Interests:

- **Customer:** Smooth ordering experience, accurate pricing
- **Staff:** Clear order details for fulfillment
- **Dining Services:** Accurate sales and revenue tracking
- **UI/UX:** Intuitive cart and checkout flow

Accessibility Expert: Compliance with accessibility standards
Preconditions: Customer logged in; items available
Trigger: Customer clicks "Add to Cart"
Main Flow:

1. Customer browses menu and adds items to cart.
2. System updates cart with items, quantities, and subtotal.
3. Customer reviews and edits cart as needed.
4. Customer proceeds to checkout, system shows summary.
5. Customer selects tip option.
6. System calculates total, order placed.
7. Confirmation displayed; order sent to staff queue.

Subflows:

- Customers change quantities or remove items.

Alternative Flows:

- 2a. Item out of stock → System shows "unavailable".
- 5a. Custom tip invalid → System rejects negative values.
- 6a. Customer cancels → Return to cart review.

UC-02: Customer Receives Pickup Notification
Primary Actor: Customer
Stakeholders & Interests:

- **Customer:** Timely notification
- **Staff:** Smooth pickup process
- **Dining Services**: Efficient handoff
- **Developers**: Reliable notification system
- **Testers/QA:** Ensure delivery of notifications

Preconditions: Order placed; marked "ready for pickup"
Trigger: Staff marks order as ready
Main Flow:

1. System detects status change.
2. System sends notification (email, SMS, or app).
3. Customer receives details and goes to pickup.
4. Customer provides ID/order number.
5. Staff locates order, hands it to customer, marks "picked up".

Alternative Flows:

- 2a. Notification fails → System retries.
- 4a. Customer can't find number → Staff searches by name/phone.
- 5a. Missing items → Staff resolves issue.

UC-03: Customer Browses Menu and Manages Cart
Primary Actor: Customer
Stakeholders & Interests:

- **Customer:** Easy discoverability, filters, availability
- **Accessibility Expert:** WCAG compliance
- **UI/UX:** Smooth navigation, clarity
- **Developers:** Maintain reliability of cart and menu system

Preconditions: Authenticated or guest mode
Trigger: Customer opens menu
Main Flow:

1. System displays accessible menu with categories.
2. Customer views details, filters, and selects options.
3. Customer adds/removes items; system updates cart live.
4. System calculates subtotal, tax estimate, tip preview.
5. Customer proceeds to checkout.

Alternative Flows:

- 2a. Out-of-stock → Add button disabled with explanation.

UC-04: Customer Checkout with Tax and Tip
Primary Actor: Customer
Stakeholders & Interests:

- **Customer:** Clear totals, transparency
- **Dining Services:** Tax compliance
- **Wolfcafe Owner:** Accurate revenue tracking
- **Regulators/Legal:** Compliance with taxation laws
- **Testers/QA:** Validate payment and calculations

Preconditions: Cart has items; tax configured
Trigger: Customer clicks checkout
Main Flow:

1. System shows summary and subtotal.
2. System calculates tax, customer selects tip.
3. System computes final total.
4. Customer enters payment and pickup info.
5. System authorizes payment, creates pending order.
6. Order confirmed and sent to staff queue.

Alternative Flows:

- 5a. Payment failure → Retry with error summary.
- 2a. Invalid custom tip → Error and reset.

UC-05: Customer Customizes Tip Amount
Primary Actor: Customer
Stakeholders & Interests:

- **Customer:** Clear totals, transparency
- **Dining Services:** Tax compliance
- **Wolfcafe Owner:** Accurate revenue tracking
- **Regulators/Legal:** Compliance with taxation laws
- **Testers/QA:** Validate payment and calculations

Accessibility Expert: Inclusive input validation
Preconditions: Items in cart; checkout started
Trigger: Customer reaches tip step
Main Flow:

1. System shows tip options.
2. Customer selects "Custom" and enters amount.
3. System validates entry, updates total.
4. Order proceeds to confirmation.

Alternative Flows:

- 2a. Negative tip → Error message.
- 2b. Very high tip → Confirmation dialog.
- 2c. Customer changes mind → Return to tip options.

UC-06: Staff Views and Fulfills Customer Order
Primary Actor: Staff
Stakeholders & Interests:

- **Staff:** Clear queue and order details
- **Customer:** Timely preparation and pickup notification
- **Dining Services:** Efficient processing
- **Wolfcafe Owner:** Reliable operations and fulfillment
- **Food Service Specialist:** Good preparation practices

Preconditions: Staff logged in; pending orders exist
Trigger: Staff opens order fulfillment interface
Main Flow:

1. System displays pending orders.
2. Staff selects order, views details.
3. Staff marks "in progress", then "ready for pickup".
4. System notifies customer.
5. Customer arrives, staff confirms identity, marks "completed".

Alternative Flows:

- 2a. Missing ingredients → Staff marks unavailable, notifies customer.
- 4a. Notification fails → Staff contacts customer manually.
- 5a. Customer late → Staff calls or sets aside order.

UC-07: Staff Creates a New Menu Item
Primary Actor: Staff
Stakeholders & Interests:

- **Staff:** Needs to manage the menu efficiently
- **Customer:** Wants to see an up-to-date and accurate menu
- **Dining Services:** Efficient processing
- **Wolfcafe Owner:** Reliable operations and fulfillment
- **Food Service Specialist:** Cares that the item and its recipe are viable

Preconditions: The Staff user is authenticated; ingredients required for the new item's recipe exist in the system
Trigger: The Staff user decides to add a new product (e.g., "Iced Caramel Latte") to the menu
Main Flow:

1. The system displays pending orders.
2. Staff selects order, views details.
3. Staff marks "in progress", then "ready for pickup".
4. The system notifies customers.
5. Customer arrives, staff confirms identity, marks "completed".

Alternative Flows:

- 3a. Missing Information → Name or Price left blank → System shows error and prevents saving.
- 1a. Permission Denied → Admin or Customer tries to access → System denies access.

UC-08: Staff adds Inventory
Primary Actor: Staff
Stakeholders & Interests:

- **Staff:** Needs to keep inventory levels accurate to prevent ordering of unavailable items
- **Wolfcare Owner:** Wants to track inventory for cost management
- **UI/UX:** Intuitive cart and checkout flow

Preconditions: The Staff user is authenticated; ingredients are already defined in the system
Trigger: A new shipment of supplies has arrived, and the Staff user needs to update the to invent the update counts
Main Flow:

1. Staff finds ingredients on the Inventory Management page and chooses to add stock.
2. Staff enters quantity being added and confirmed.
3. System validates amount is positive number, adds to existing stock and updates UI.

Alternative Flows:

- 2a. Invalid Quantity**:** Quantity is zero, negative, or not a number → System shows error and does not update inventory.

UC-09: Admin Manages Staff User Account
Primary Actor: Admin

Stakeholders & Interests:
- **Admin:** Needs to control system access securely
- **Staff:** Clear queue and order details
- **Wolfcafe Owner:** Wants to ensure only authorized personnel can access staff functions

Preconditions: Staff logged in; pending orders exist
Trigger: Staff opens order fulfillment interface
Main Flow:
1. Admin selects “Manage Staff” from the dashboard.
2. System displays existing Staff user list.
3. Admin chooses "Add New Staff" (or "Edit"/"Delete" existing).
4. System presents form for Staff details.
5. Admin fills required information and submits.
6. System validates input and creates/updates/deletes account.
7. System logs action and displays success message.

Alternative Flows:
- 2a. Missing ingredients → Staff marks unavailable, notifies customer.
- 4a. Notification fails → Staff contacts customer manually.
- 5a. Customer late → Staff calls or sets aside order.

UC-10: Admin Sets Sales Tax Rate
Primary Actor: Admin
Stakeholders & Interests:
- **Admin:** Needs to ensure cafe compliance with state tax regulations
- **WolfCafe Owner:** Wants accurate financial accounting
- **Customer:** Is affected by tax calculation on final bill
- **Regulator:** Requires compliance with local tax laws

Preconditions: The Admin is authenticated; system has default or previously set tax rate
Trigger: The Admin accesses system's financial settings to update sales tax percentage
Main Flow:
1. Admin navigates to "System Settings" → "Financial".
2. System displays current sales tax rate.
3. Admin enters new tax rate value and clicks "Save".
4. System validates input is valid, non-negative number.
5. System updates global tax rate and displays confirmation.

Alternative/Exception Flows:
- 4a. Invalid Input: Input not a number or negative → System shows inline error and prevents saving.
- 1a. Permission Denied: Staff or Customer user attempts access → System redirects with permission error.