CSC 510 - Project 1a1

Christopher Lee, Kunj Patel, Fan-Sheng Chuang, and Raj Ajay Vaidya

List of Stakeholders

Primary Stakeholders

- **Admin** → Configures policies, manages staff accounts, and ensures compliance with both internal and external rules.
- **Staff** → Prepares orders, maintains kitchen workflows, and communicates with customers during fulfillment.
- **Customer** → Places and customizes orders, tracks delivery or pickup status, and provides feedback.

Secondary Stakeholders

- **Delivery Personnel** → Ensures orders reach customers offsite and provides proof of delivery.
- **Developers** → Extend and maintain the system, balancing performance and new features.
- **Quality Engineers** → Conduct testing, detect regressions, and validate new releases.
- **Teaching Staff** → Function as product owners, ensuring alignment with academic goals.
- **IT Security Officers** → Enforce identity verification, encryption, and data governance.
- **Payment Gateways** → Enable secure transaction processing (e.g., card, wallet).

External Stakeholders

- **Vendors/Suppliers** → Provide ingredients and consumables to the cafe.
- **Accessibility Advocates** → Ensure equitable user experience across abilities.
- **Auditors and Regulators** → Enforce compliance with tax, labor, and privacy law.

Stakeholder Biases

Stakeholder A	Stakeholder B	Clash	Example
Customer	Staff	Customers desire near-instant service, while staff need sustainable preparation speed.	Long special instructions slow down kitchen throughput.
Admin	Developer	Admins want rapid customization and dashboards; developers prefer stability and minimal technical debt.	Admin requests frequent UI tweaks that break existing tests.
Delivery	Customer	Couriers may batch routes for efficiency; customers expect precise timing.	Delivery driver groups orders to save fuel, customer complains about late order.
Security Officers	Customers	Security enforces multi-factor logins; customers want one-click checkout.	MFA delays login on mobile app compared to single tap checkout.
Regulators	Business	Compliance bodies demand strict standards; business managers prioritize low operational costs.	Regulator mandates allergen labeling that requires expensive software changes.

Prompt Crafting: Zero-Shot vs Careful Prompting

Zero-Shot Prompting

Zero-shot involves asking the LLM a question without constraints. For example, "List stakeholders in WolfCafe." This may produce a mix of relevant and irrelevant answers. It is fast but inconsistent.

Careful Prompting

Careful prompting provides structured formats, role examples, and style constraints. For example, "List WolfCafe stakeholders in the form Role → short responsibility." This produces more consistent and useful content, albeit at the cost of creativity. In practice, careful prompting yields professional documentation, while zero-shot is useful for brainstorming.

Use Cases

Use Case 1: Multi-Device Login

Preconditions: User has an existing account with at least one active session.

Main Flow:

1. Customer launches the WolfCafe mobile app.
2. Customer enters username and password on the login screen.
3. System validates credentials against the identity service.
4. System checks for existing active sessions on other devices.
5. If a desktop session is active, the system generates a push notification to that device.
6. The desktop client displays a message: “A new device is attempting to log in.”
7. Customer confirms via the desktop notification that the session coexistence is acceptable.
8. System updates the account record to allow multiple concurrent sessions.
9. Mobile session is initiated; both devices now maintain synchronized state.
10. Order history, favorites, and cart contents are synchronized across devices.

Subflows:

- **[S1]** Two-factor authentication prompts the user to enter a code sent via SMS or authenticator app.
- **[S2]** If one of the sessions is idle for more than a set time, system automatically logs out that session but keeps others active.
- **[S3]** Customer may choose “log out other sessions” to terminate all concurrent logins except the current device.

Alternative Flows:

- **[E1]** If maximum device count is exceeded, system denies login with a message.
- **[E2]** If notification is not acknowledged within 2 minutes, login attempt fails.
- **[E3]** Suspicious location triggers fraud detection and blocks login.

Notes: Multi-device login enhances usability but requires careful management of security and synchronization.

Use Case 2: Guest Checkout

Preconditions: User is not logged in.

Main Flow:

1. Customer browses menu and adds items to the cart.
2. System saves cart under a temporary guest session.

3. Customer selects “Checkout as Guest.”
4. System requests minimal details (name, email, phone).
5. Customer enters payment method.
6. System validates payment format and applies fraud checks.
7. System creates a temporary guest order record.
8. Order confirmed; receipt sent via email/SMS.

Subflows:

- [**S1**] Guest switches to account creation mid-process; cart migrates.
- [**S2**] Guest chooses alternative payment (e.g., wallet).
- [**S3**] System offers to save payment info for future account.

Alternative Flows:

- [**E1**] Payment fails → retry or switch method.
- [**E2**] Fields missing → inline validation prompts correction.
- [**E3**] Fraud detected → order blocked, support notified.

Notes: Guest checkout improves adoption but increases risk of fraud.

Use Case 3: Allergy Warnings

Preconditions: Menu items contain allergy metadata.

Main Flow:

1. Customer selects a menu item.
2. System retrieves allergen metadata linked to the item.
3. The interface highlights allergens (e.g., dairy, nuts, gluten).
4. If customer has an allergy profile, system cross-checks with it.
5. System displays a warning message: “This dish contains peanuts.”
6. Customer acknowledges the warning and decides to proceed or cancel.
7. System records the acknowledgement for liability tracking.

Subflows:

- [**S1**] Customer adds allergy preferences in settings, which are used for all future orders.

- **[S2]** System suggests alternative menu items that do not conflict with allergies.
- **[S3]** Customer shares allergen information with support for special preparation requests.

Alternative Flows:

- **[E1]** If allergen data is missing, the system displays "Allergen info not available."
- **[E2]** If allergens conflict with saved profile, checkout is temporarily blocked until acknowledgment.
- **[E3]** If allergen metadata is inconsistent, the system escalates to the admin dashboard.

Notes: Allergy handling is crucial for customer safety and legal compliance.

Use Case 4: Tip Suggestions

Preconditions: Payment interface active.

Main Flow:

1. Customer navigates to the checkout screen.
2. System displays a section for tipping options.
3. Suggested tip percentages (10%, 15%, 20%) are shown by default.
4. Customer selects one of the percentages or customizes the tip amount.
5. Tip value is added to the total payment.
6. System updates the receipt preview.
7. Customer confirms payment including tip.

Subflows:

- **[S1]** Customer chooses "no tip" and proceeds.
- **[S2]** Customer enters custom tip in dollar value instead of percentage.
- **[S3]** System reminds the customer if no tip was selected during peak hours.

Alternative Flows:

- **[E1]** Invalid input (e.g., letters in tip field) triggers validation error.
- **[E2]** If customer balance is insufficient, system prompts to reduce tip or change payment.

Notes: Tip suggestions encourage fair staff compensation while allowing customer flexibility.

Use Case 5: Order Handoff Confirmation

Preconditions: Order prepared by staff.

Main Flow:

1. Staff marks the order as “ready.”
2. Customer approaches pickup counter and scans QR code.
3. System cross-checks order ID with QR code.
4. System displays confirmation message on staff dashboard.
5. Staff hands over order to customer.
6. Order status updated to “completed.”

Subflows:

- [**S1**] Curbside pickup triggers location-based notification for staff.
- [**S2**] Locker pickup assigns a locker number and code.
- [**S3**] Staff manually verifies customer ID if scanner malfunctions.

Alternative Flows:

- [**E1**] Wrong code scanned → alert displayed, manual verification required.
- [**E2**] Customer arrives too late → order disposed or refunded.
- [**E3**] Customer disputes missing item → triggers support case.

Notes: Order handoff confirmation reduces fraud and ensures accountability.

Use Case 6: Scheduled Maintenance Notices

Preconditions: Developer/admin plans downtime.

Main Flow:

1. Admin schedules a maintenance window in the backend.
2. System records start and end times.
3. Banner displayed on website and app 24 hours in advance.
4. Customers warned that ordering will be disabled during maintenance.
5. At maintenance start, ordering and payments are locked.

6. After maintenance, system services resume automatically.

Subflows:

- **[S1]** Admin posts detailed patch notes alongside notice.
- **[S2]** Customers subscribed to updates get an email alert.
- **[S3]** Urgent maintenance overrides pre-scheduled banner timing.

Alternative Flows:

- **[E1]** Emergency outage requires immediate shutdown.
- **[E2]** Maintenance exceeds expected duration → banner auto-extends.

Notes: Maintenance transparency builds trust and prevents user frustration.

Use Case 7: Digital Receipt Archival

Preconditions: Completed order exists.

Main Flow:

1. After payment, system generates digital receipt.
2. Receipt stored in customer's profile.
3. Customer receives receipt via email.
4. Receipts organized chronologically in app.
5. Customer can download receipts as PDF.
6. Search function allows receipts retrieval by date or order ID.

Subflows:

- **[S1]** Customer tags receipts with labels (e.g., "business expense").
- **[S2]** Integration with expense tracking software exports receipts automatically.
- **[S3]** Archival policy auto-deletes receipts after 2 years unless marked important.

Alternative Flows:

- **[E1]** Customer email bounces → fallback notification in app.
- **[E2]** PDF generation fails → simple text fallback sent.

Notes: Digital receipts improve record keeping and support business customers.

Use Case 8: Dynamic Pricing

Preconditions: Admin has pricing rules defined.

Main Flow:

1. System monitors demand in real-time.
2. At peak load, triggers price adjustment rules.
3. Prices displayed to customer are updated instantly.
4. Order total reflects new prices.
5. System records rule execution for auditing.

Subflows:

- **[S1]** Admin defines happy hour discounts.
- **[S2]** Seasonal pricing adjusts automatically during holidays.
- **[S3]** VIP customers exempted from dynamic price increases.

Alternative Flows:

- **[E1]** Conflicting rules → system defaults to base price.
- **[E2]** Price spike exceeds cap → rollback executed.

Notes: Dynamic pricing balances demand but must avoid customer alienation.

Use Case 9: Order Sharing

Preconditions: Customer account exists.

Main Flow:

1. Customer selects past order.
2. System generates shareable link or QR code.
3. Customer sends link to a friend.
4. Friend opens link and sees order summary.
5. Friend clones and customizes order.
6. New order placed under friend's account.

Subflows:

- **[S1]** Group ordering mode allows multiple customers to add items to a shared cart.

- **[S2]** Friends can comment on shared order before finalizing.
- **[S3]** System prevents expired or deleted orders from being shared.

Alternative Flows:

- **[E1]** Link expired → customer prompted to re-generate.
- **[E2]** Recipient lacks account → system suggests guest checkout.

Notes: Order sharing adds social value and drives retention.

Use Case 10: Carbon Footprint Report

Preconditions: System stores item sourcing metadata.

Main Flow:

1. Customer requests sustainability report at checkout.
2. System aggregates sourcing data (origin, transport mode).
3. Emissions values assigned to each item.
4. Total carbon footprint calculated for order.
5. Report displayed in comparison with average meal footprint.
6. Customer sees suggestions for lower-carbon alternatives.

Subflows:

- **[S1]** Admin uploads verified carbon data from suppliers.
- **[S2]** System integrates third-party sustainability API.
- **[S3]** Customer opts to donate to carbon offset projects at checkout.

Alternative Flows:

- **[E1]** Missing data → partial report shown with disclaimer.
- **[E2]** API failure → cached averages used.