Proj1a1 - Group 4

Q1) List all stakeholders.

A1) Customers, Staff (Manager, Barista), Administrators, Developers, Government/Regulators, Payment providers

Follow up questions

a) Can you summarize in a tabular form for easy visualisation.

Stakeholder	Type	Primary Interests / Responsibilities	Concerns / Risks
Customers	Direct User	Place orders, add multiple items, add tips, receive notifications when order is ready. Option for anonymous ordering.	Friction if forced to log in; service speed and ease of use.
Staff	Direct User	Create new items/recipes, add to inventory, view/fulfill orders, review order history for inventory planning.	Complexity if roles are split; need flexibility in small cafés.
Manager (subset of Staff)	Direct User	Maintain and manage inventory, oversee recipes.	Risk of being blocked from helping with orders if roles too rigid.
Barista (subset of Staff)	Direct User	Fulfill customer orders, create new recipes.	Lack of permission to adjust inventory in emergencies.
Administrators (Admins)	Direct User	Manage staff & customer accounts, set sales tax rates, oversee permissions.	Anonymous orders bypass account management; misconfigured tax compliance.
Developers (Student Team)	Indirect User	Build features, write documentation (Developers' Guide, Users' Guide), implement/test/refactor system.	Balance between feature development and technical debt/documentation.
Government / Regulators	Indirect Stakeholder	Ensure compliance with financial regulations (sales tax, NC food tax).	Risk of non-compliance → legal/financial penalties.
Payment Providers	Indirect Stakeholder	Process transactions (orders with tips + sales tax).	Must handle secure, reliable payment flow; downtime risks affect all.

b) Are there any obscure stakeholders in this system (e.g., marketing team, customer support, third-party integrations)? List them and explain their roles briefly.

A 1b)

01. Original "CoffeeMaker" Developers/Maintainers
    a. The student team must integrate and potentially refactor existing CoffeeMaker functionality. The original developers are stakeholders because their design choices, code quality, and documentation directly shape the new team's work.
02. Third-Party Payment Processors
    a. Although not explicitly implemented, calculating order totals with tax and tips implies integration with real-world payment gateways (e.g., Stripe, Square, PayPal). These providers are stakeholders since they impose technical requirements on the system.
03. Third-Party Development Tool & Platform Providers
    a. The project relies on external tools that influence development such as GitHub, Testing Libraries, Code Coverage & Badge Services.

c) For each stakeholder, summarize their primary goals and what they want from the food delivery system.

A 1c) Direct Users are:

01. Customers:
    a. Want to order items quickly, including multiples of the same item.
    b. Expect clear pricing with sales tax and tipping options (15%, 20%, 25%, or custom).
    c. Need real-time updates on order status and pickup readiness.
    d. Prefer optional account-less ordering.

02. Staff:
    a. Fulfill orders by viewing and selecting from incoming lists.
    b. Manage offerings by creating new items/recipes.
    c. Track and update inventory levels.
    d. Review order history for revenue analysis and inventory planning.
03. Administrators:
    a. Manage staff accounts (create, edit, delete).
    b. Manage customer accounts (edit, delete).
    c. Control system-wide financial settings (e.g., sales tax rate).

Indirect Stakeholders are:

01. Developers (Student Team):
    a. Build the system to meet requirements and grading criteria.
    b. Deliver backend + frontend functionality with sufficient test coverage (≥70%).
    c. Produce complete documentation (Developers' Guide, Users' Guide).
02. Original "CoffeeMaker" Developers:
    a. Not directly involved, but their prior design/code must be successfully integrated into WolfCafe.
03. Third-Party Payment Processors:
    a. Not explicitly included, but implied: provide secure transaction handling for totals, tax, and tips.

Q2) Identify stakeholder biases
A2)

Developers vs. Customers — Technical Rigor vs. Usability	**Administrators vs. Customers — Security vs. Convenience**	**Staff vs. Developers — Operational Efficiency vs. Scope Limitations**	**Customers vs. Staff — Personalization vs. Fulfillment Speed**
Developers must prioritize backend testing, documentation, and grading criteria.	Administrators want account-based control over users.	Staff want analytics-friendly tools (visualizations, insights).	Customers value order customization (special requests, add-ons).
Customers want an intuitive, visually appealing ordering interface.	Customers prefer quick, sometimes anonymous ordering.	Developers may only build a basic table view to meet minimum requirements.	Staff need standardized items for fast, error-free fulfillment.
Conflict: Focus on backend deliverables may leave customers with a clunky UI.	Conflict: Account enforcement slows down customers, while anonymous ordering	Conflict: Staff lose out on efficient decision-making tools due to scope	Conflict: Personalization slows staff efficiency and

	weakens admin error.	constraints.	increases mistakes.

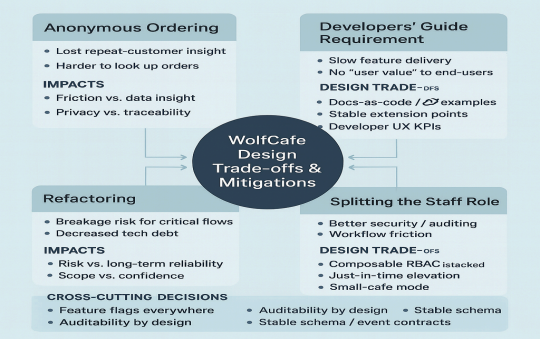
Follow up questions:

Q 2a) Give examples of requirements that might be important to one stakeholder but irrelevant or harmful to another.

A 2a)

Anonymous Ordering	Developers' Guide Requirement	Refactoring Option	Splitting the Staff Role
Important to: Customers (fast, no-login orders).	Important to: Developers & Teaching Staff (graded deliverable).	Important to: Developers (better maintainability, cleaner design).	Important to: Administrators (clear separation of duties, security).
Harmful to: Administrators (loses account control), Staff (weaker order history & sales tracking).	Irrelevant to: Customers & Staff (time spent on docs means fewer user-facing features).	Harmful to: Customers & Staff (risk of bugs, no immediate feature gains).	Harmful to: Staff in small cafes (less flexibility when employees need to cover multiple tasks).

Q2 b) Suggest how these stakeholder conflicts could impact system design or trade-offs, visually.



Q3) Comment on prompt crafting (Compare zero-shot prompting to careful prompting)

A3)

Zero-Shot Prompting	Careful Prompting
Give a broad task with little context.	Provide structured, detailed instructions.
Output: quick but inconsistent, may miss depth or style.	Output: polished, consistent, closer to expectations.
Good for: brainstorming, idea generation.	Good for: professional docs, visuals, reports.
Trade-off: speed vs. precision.	Trade-off: upfront effort vs. quality.

Key Insight:

shot = exploration, Careful prompting = execution

Q4) Write at least 10 use cases (≈5 pages total), each with: Preconditions, Main Flow, Subflows, Alternative Flows.

A4)

01. Use Case: Create Staff User

a. **Actor:** Administrator
b. **Description:** An Administrator creates a new user account with the "Staff" role, enabling that person to perform staff duties.
c. **Preconditions:**
    i. The user is logged in as an Administrator.
d. **Main Flow:**
    i. The Administrator navigates to the user management section.
    ii. The Administrator selects the option to create a new user.
    iii. The system presents a form to enter the new user's details.
    iv. The Administrator selects "Staff" from the list of available roles.
    v. The Administrator submits the form.
    vi. The system validates that the user details and that all required fields are filled.
    vii. The system creates the new Staff user account.
    viii. The system displays a success message confirming the user has been created.
e. **Alternative Flows:**
    i. **A1: Username Exists:** At step 6, system finds duplicate username, shows error, highlights field → resume at step 3.

02. Use Case: Delete Customer User

a. **Actor:** Administrator
b. **Description:** An Administrator permanently removes a customer's account from the system.
c. **Preconditions:**
    i. The user is logged in as an Administrator.
    ii. There is at least one existing customer account in the system.
d. **Main Flow:**
    i. The Administrator navigates to the user management section.
    ii. The Administrator searches for or selects the customer account to be deleted.
    iii. The Administrator selects the "Delete" option for that user.
    iv. The system displays a confirmation prompt to prevent accidental deletion.
    v. The Administrator confirms the deletion.
    vi. The system permanently removes the customer's account and associated data.
    vii. The system displays a success message confirming the deletion.
e. **Alternative Flows:**

i. **A1: Deletion is Canceled:** At step 5, admin cancels deletion, system closes prompt → no changes, use case ends.

03. Use Case: Set Sales Tax Rate

a. **Actor:** Administrator
b. **Description:** An Administrator sets the system-wide sales tax rate applied to all orders.
c. **Preconditions:**
   i. The user is logged in as an Administrator.
d. **Main Flow:**
   i. The Administrator navigates to the system settings or configuration panel.
   ii. The Administrator locates the "Sales Tax Rate" setting.
   iii. The Administrator enters a new numerical value for the tax rate percentage.
   iv. The Administrator saves the new setting.
   v. The system validates the input is a valid, non-negative number.
   vi. The system updates the sales tax rate.
   vii. The system displays a success message.
e. **Alternative Flows:**
   i. **A1: Invalid Input:** At step 5, system detects invalid input (text or negative), shows error → resume at step 3.

04. Use Case: Create New Menu Item

a. **Actor:** Staff
b. **Description:** A staff member adds a new item or recipe to the menu, making it available for customers to order.
c. **Preconditions:**
   i. The user is logged in with a "Staff" role.
d. **Main Flow:**
   i. The Staff member navigates to the menu or inventory management section.
   ii. The Staff member selects the option to create a new item/recipe.
   iii. The system presents a form for the new item's details (e.g., name, price).
   iv. The Staff member fills out the required information and submits the form.
   v. The system validates the input (e.g., price is a positive number).
   vi. The system creates the new menu item with an initial inventory of zero.
   vii. The system displays a success message.
e. **Alternative Flows:**
   i. **A1: Invalid Price:** At step 5, system detects non-positive price, shows error → resume at step 3.

05. Use Case: Add Inventory

a. **Actor:** Staff
b. **Description:** A staff member increases the available stock for a specific menu item.
c. **Preconditions:**

i. The user is logged in with a "Staff" role.
ii. At least one menu item/recipe exists in the system.

d. **Main Flow:**
i. The Staff member navigates to the inventory management section.
ii. The Staff member selects the item to which they want to add inventory.
iii. The system displays the current inventory count and an input field to add more.
iv. The Staff member enters the quantity to add.
v. The Staff member confirms the addition.
vi. The system validates the input is a positive integer.
vii. The system updates the inventory count for the selected item.
viii. The system displays a success message showing the new total inventory.

e. **Alternative Flows:**
i. **A1: Invalid Quantity:** At step 6, system detects non-positive integer, shows error → resume at step 4.

06. Use Case: Fulfill Order

a. **Actor:** Staff
b. **Description:** A staff member views the queue of pending orders and marks one as completed and ready for pickup.
c. **Preconditions:**
i. The user is logged in with a "Staff" role.
ii. At least one customer has placed an order that is pending fulfillment.
d. **Main Flow:**
i. The Staff member navigates to the order fulfillment screen.
ii. The system displays a list of all active, unfulfilled orders.
iii. The Staff member selects an order from the list to fulfill.
iv. The system displays the details of the selected order.
v. After preparing the order, the Staff member selects the "Fulfill Order" button.
vi. The system updates the order's status to "Fulfilled".
vii. The system removes the order from the active queue and sends a notification to the customer's display.
viii. The use case ends.
e. **Alternative Flows:**
i. **A1: Order Canceled by Customer (if feature exists):** At step 3, system finds order canceled, shows message, removes from list → resume at step 2.

07. Use Case: Place Order

a. **Actor:** Customer
b. **Description:** A customer selects items, adds them to a cart, adds a tip, and confirms their order for pickup.
c. **Preconditions:**

i. The Customer is logged into the system.
ii. At least one item is available for sale with sufficient inventory.
iii. The Administrator has set a sales tax rate.

d. **Main Flow:**
i. The Customer browses the available menu items.
ii. The Customer selects an item and adds it to their order.
iii. The system updates the order total, including sales tax.
iv. The Customer repeats steps 2-3 for all desired items.
v. The Customer proceeds to finalize the order.
vi. The system displays the final total and prompts for a tip (15%, 20%, or custom).
vii. The Customer selects a tip option.
viii. The Customer confirms and submits the order.
ix. The system validates inventory, creates the order, and places it in the fulfillment queue.
x. The system displays a confirmation screen to the Customer.

e. **Subflows:**
i. **S1: Add Custom Tip: S1:** At step 7, customer selects custom tip, enters amount → system updates total → continue at step 8.

f. **Alternative Flows:**
i. **A1: Insufficient Inventory:** At step 2, system finds item out of stock, shows message → resume at step 1.

08. Use Case: Monitor Order Status

a. **Actor:** Customer
b. **Description:** A customer checks the status of their placed order to see when it is ready for pickup.
c. **Preconditions:**
i. The Customer is logged in.
ii. The Customer has already placed an order that is pending fulfillment.

d. **Main Flow:**
i. The Customer navigates to their order history or views their active order display.
ii. The system shows the current status of the order (e.g., "In Progress").
iii. When a Staff member fulfills the order, the system automatically updates the status on the Customer's display to "Fulfilled" or "Ready for Pickup".
iv. The use case ends.

e. **Alternative Flows:** There are no significant alternative flows for this passive use case.

09. Use Case: View Order History

a. **Actor:** Staff

b. **Description:** A staff member views a log of past completed orders for business analysis.
c. **Preconditions:**
    i. The user is logged in with a "Staff" role.
    ii. At least one order has been fulfilled in the past.
d. **Main Flow:**
    i. The Staff member navigates to the "Order History" section.
    ii. The system displays a list or table of previously fulfilled orders, showing details like items purchased, total cost, and timestamp.
    iii. The Staff member can optionally filter or sort the data (if supported).
    iv. The Staff member reviews the information.
    v. The use case ends when the Staff member navigates away.
e. **Subflows:**
    i. **S1: View Order Details:** At step 2, staff selects an order → system shows detailed view.
f. **Alternative Flows:**
    i. **A1: No Order History:** At step 2, system finds no orders, shows message → use case ends.

10. Use Case: Place Anonymous Order

a. **Actor:** Customer (not logged in)
b. **Description:** A user who does not have an account or is not logged in places an order.
c. **Preconditions:**
    i. The user is not logged into the system.
    ii. Items are available for sale with sufficient inventory.
d. **Main Flow:**
    i. The anonymous user browses the menu items.
    ii. The user adds items to a temporary order/cart.
    iii. The system updates the order total, including sales tax.
    iv. The user proceeds to finalize the order.
    v. The system prompts for some information to identify the order.
    vi. The user provides the required information.
    vii. The user selects a tip option and confirms the order.
    viii. The system creates the order and places it in the fulfillment queue, associated with the identifier provided in step 6.
    ix. The system displays a confirmation screen with an order number that the user can use to track their order.
e. **Alternative Flows:**
    i. **A1: Required Information Not Provided:** At step 6, system detects missing name/ID, shows error → resume at step 5.