CSC 326 Project 1a1: WolfCafe Food Delivery System Analysis

Project: 1a1

Due: Tuesday, Sept 2, noon

Objective: Problem familiarization for WolfCafe food delivery system

---

1. Create List of Stakeholders

Primary Stakeholders

- **Administrator** - System management, user account creation, tax rate configuration.
- **Staff** - Order fulfillment, inventory management, food preparation (includes Baristas, Kitchen Staff, Managers).
- **Customer** - Order placement, payment processing, order pickup.

Secondary Stakeholders

- **Teaching Staff (Instructors/TAs)** - Requirements definition, project evaluation, milestone approval.
- **Development Team** - System implementation, feature development, code maintenance.
- **Payment Processors** - Transaction handling, tip processing, refund management.
- **Food Suppliers** - Inventory provision, supply chain management.
- **IT Support** - System maintenance, technical troubleshooting.

Regulatory Stakeholders

- **Tax Authorities (NC Department of Revenue)** - Sales tax compliance (2.0% NC food tax).
- **Food Safety Inspectors** - Health code compliance, safety regulations.
- **Data Protection Authorities** - Privacy policy compliance, data security.
- **Accessibility Regulators** - ADA compliance requirements.

External Stakeholders

- **System Integrator** - CoffeeMaker to WolfCafe migration specialist.
- **QA/Test Engineers** - Quality assurance, automated testing.
- **Environmental Groups** - Sustainability concerns, packaging waste reduction.
- **Local Community** - Noise impact, traffic patterns, local business effects.
- **Competitors (GrubHub, DoorDash)** - Market influence, feature expectations.
- **End Users** - Broader university community beyond immediate project scope.

---

2. Identify Stakeholder Biases

5 Ways Stakeholder Needs Clash or Are Irrelevant to Others:

1. **System Security vs. User Convenience**.
    - **Data Protection Authorities** mandate extensive authentication, data encryption, and audit trails.
    - **Customers** want quick, frictionless ordering and potentially anonymous purchases.
    - **Clash**: Regulatory compliance directly opposes user experience, creating tension between legal safety and usability.

2. **Environmental Sustainability vs. Operational Efficiency**.
    - **Environmental Groups** push for biodegradable packaging, minimal waste, and sustainable sourcing.
    - **Staff/Management** prioritize cost-effective materials, fast service, and operational simplicity.
    - **Clash**: Eco-friendly options often cost more and take more time to handle, which conflicts with efficiency metrics and budget constraints.
3. **Feature Innovation vs. System Stability**.
    - **Development Team** wants to implement new features, refactoring improvements, and experimental technologies.
    - **Teaching Staff/Administrators** need a reliable system that meets core requirements within the academic timeline.
    - **Clash**: Innovation introduces risk and complexity that conflicts with project stability requirements and grading deadlines.
4. **Revenue Optimization vs. Customer Value**.
    - **Payment Processors** aim to maximize transaction fees, tip processing, and service charges.
    - **Customers** seek the lowest total costs, transparent pricing, and simple payment options.
    - **Irrelevance**: Payment processor profit margins are irrelevant to customer satisfaction, creating an inherent conflict in system design priorities.
5. **Individual Workflow Efficiency vs. Compliance Oversight**.
    - **Staff** wants streamlined personal workflows, minimal documentation, and autonomous decision-making.
    - **Regulatory Bodies** require detailed audit trails, standardized processes, and oversight mechanisms.
    - **Irrelevance**: The staff's desire for efficiency by reducing documentation is irrelevant to regulators, who need comprehensive compliance tracking, creating operational friction.

Stakeholder	Primary Bias	Competing Stakeholder	Conflicting Bias	Example Manifestation	Impact on WolfCafe	Resolution Strategy
Customer	Convenience-First	Data Protection Authorities	Privacy-Maximization	Customers want one-click ordering vs. regulators requiring multi-step authentication	Simple checkout vs. security compliance	Implement optional express checkout with risk assessment
Staff	Efficiency-Optimization	Environmental Groups	Sustainability Focus	Staff prefers disposable packaging for speed vs. eco-friendly reusable containers	Fast service vs. environmental responsibility	Introduce eco-incentives and efficiency training

Administrator	Cost-Minimization	Food Safety Inspectors	Quality-Maximization	Admin cuts costs on premium ingredients vs. inspectors requiring highest safety standards	Budget constraints vs. compliance costs	Tier safety investments: non-negotiable basics, optional premiums
Development Team	Innovation-Driven	Teaching Staff	Deadline-Adherence	Developers want cutting-edge features vs. instructors needing core requirements completed on time	Extra credit features vs. project completion	Time-box innovation: core first, then enhancements
Payment Processors	Revenue-Maximization	Customers	Value-Optimization	Processors push complex fee structures vs. customers wanting transparent, low costs	Transaction fees vs. customer satisfaction	Negotiate bulk rates, pass savings to customers

3. Comment on Prompt Crafting: Zero-Shot vs. Careful Prompting

Zero-Shot Prompting

- **Definition**: Providing an AI with a task request without examples, context, or specific formatting guidance.
- **Example**: "List stakeholders for a food delivery system".
- **Results**:
  - Produces generic, surface-level responses.
  - Misses project-specific context (WolfCafe, academic requirements).
  - Inconsistent formatting and structure.
  - May overlook domain-specific stakeholders.
  - Limited depth of analysis.

Careful Prompting

- **Definition**: Structured prompting with specific context, examples, formatting requirements, and iterative refinement.
- **Example**: "For the WolfCafe food delivery system (similar to GrubHub) being developed as an academic project integrating CoffeeMaker functionality, identify all stakeholders including direct users, regulatory bodies, and indirect community impacts. Categorize them as Primary, Secondary, Regulatory, and External. For each category, explain their specific role and system interactions".
- **Results**:
  - Comprehensive, contextually relevant responses.
  - Professionally formatted, structured output.
  - Domain-specific insights and terminology.
  - Addresses project-specific requirements.
  - Higher quality, actionable deliverables.

Comparison Conclusion

Careful prompting significantly outperforms zero-shot approaches for academic projects that require specific deliverable formats, domain expertise, and comprehensive analysis. The effort invested in prompt engineering directly correlates with the quality of the output and the fulfillment of project requirements.

---

4. Write at least 10 Use Cases

Use Case 1: Administrator Sets Sales Tax Rate

Preconditions
The administrator is logged in with system privileges, the configuration module is operational, and a default tax rate (2.0%) exists.

Main Flow

1. Administrator accesses the administrative dashboard and selects "Tax Settings."
2. System displays current tax rate and last update timestamp.
3. Administrator clicks "Edit Tax Rate" and enters a new value.
4. The system validates input format and ensures the percentage is within an acceptable range.
5. Administrator reviews the change summary and confirms update.
6. System updates the tax rate across all active calculations, logs the change, and displays confirmation.

Subflows & Alternatives

- Invalid input → error prompt requests correction.
- Database error → system reverts to old rate and notifies IT support.
- Administrator may preview impact on sample orders before applying.

Outcome
All transactions reflect the new tax rate with an audit trail recorded for compliance.

Use Case 2: Staff Creates New Menu Item

Preconditions
Staff is authenticated with menu management permissions. The inventory database is available, and the menu interface is operational.

Main Flow

1. Staff accesses "Menu Management" and selects "Add New Item."
2. System provides a form requiring name, description, category, and price.
3. Staff fills in details, associates ingredients, and sets availability.
4. Optional: staff uploads an image.
5. System validates required fields and confirms item creation.
6. The new item appears in the menu with a unique ID.

Subflows & Alternatives

- Missing or invalid data → system highlights errors.
- Duplicate item → system suggests modifying or editing existing record.
- Database failure → system saves as draft or requests retry.

Outcome

The menu is updated with a new item, ensuring customers see current offerings.

Use Case 3: Customer Places Multi-Item Order

Preconditions

Customer is authenticated or guest ordering is enabled. Menu items are available, and payment systems are operational.

Main Flow

1. Customer browses categories and adds multiple items to cart.
2. System displays cart summary with quantities and prices.
3. Customer reviews, adjusts items, and proceeds to checkout.
4. System calculates subtotal, tax, and tip options.
5. Customer selects payment method, confirms, and places order.
6. System processes payment, assigns order number, and provides pickup estimate.

Subflows & Alternatives

- Special instructions may be added for items.
- Payment failure → retry with another method.
- Item unavailable → customer notified, substitution offered.

Outcome

Customer receives confirmation with order tracking and pickup details.

Use Case 4: Staff Views and Fulfills Order

Preconditions

Staff has order management access, pending orders exist, and fulfillment interface is operational.

Main Flow

1. Staff logs into dashboard to view pending orders.
2. Selects an order and reviews details (items, instructions, customer info).
3. Staff begins preparation, and system updates status to "In Progress."
4. Once prepared, staff verifies contents and marks as "Ready for Pickup."
5. System notifies customer and moves order to "Ready" queue.

Subflows & Alternatives

- Low stock → customer contacted for substitution.
- Delay in preparation → revised pickup time sent.
- System error → staff logs order manually until resolved.

Outcome

Order is prepared accurately and marked ready, ensuring timely pickup.

Use Case 5: Customer Picks Up Completed Order

Preconditions

Customer has an active order marked "Ready," and staff is available at pickup location.

Main Flow

1. Customer arrives and provides order number.
2. Staff verifies order in system and confirms customer details.
3. Customer reviews items and accepts the order.
4. Staff marks order as "Picked Up."
5. System updates status to "Completed" and archives history.

Subflows & Alternatives

- Customer may use QR code for faster lookup.
- Incorrect or missing items → corrected before completion.
- System outage → pickup logged manually for later update.

Outcome

Customer successfully receives order, and feedback may be collected.

Use Case 6: Administrator Manages Staff Accounts

Preconditions

Administrator is authenticated with user management privileges. Staff roles and permissions are predefined.

Main Flow

1. Administrator accesses "Staff Management."
2. System lists active staff with roles and statuses.
3. Admin selects create, edit, or delete.
4. For create/edit, details such as name, contact, and role are updated.
5. System validates information and applies changes.
6. Confirmation is sent to staff with updated access.

Subflows & Alternatives

- Invalid details → system prompts correction.
- Active orders prevent deletion → suggests deactivation.
- Database error → changes not applied, logged for IT.

Outcome

Staff accounts remain accurate, ensuring controlled access to system features.

Use Case 7: Staff Manages Inventory Levels

Preconditions

Staff has inventory permissions, supplier and ingredient data are current, and system is operational.

Main Flow

1. Staff accesses "Inventory Management."
2. System displays ingredient quantities and status (low, normal).
3. Staff updates quantities after supplier delivery.
4. New totals and availability are recalculated.
5. Staff sets reorder thresholds for alerts.

6. System confirms and updates affected menu items.

Subflows & Alternatives

- Unrealistic quantities trigger warning.
- Low inventory items highlighted for quick action.
- Database failure → changes logged but not saved.

Outcome

Inventory remains accurate, preventing stockouts and ensuring menu reliability.

Use Case 8: System Processes Anonymous Order

Preconditions

Guest checkout is enabled, and payment processing supports anonymous transactions.

Main Flow

1. Customer browses menu without logging in.
2. Items are added to cart and checkout initiated.
3. System requests minimal details (name, contact).
4. Customer enters payment, selects tip, and confirms.
5. System processes order, assigns number, and sends confirmation.
6. Order follows normal fulfillment workflow.

Subflows & Alternatives

- Invalid contact → correction requested.
- Payment failure → retry allowed.
- High volume → extended pickup time given.

Outcome

Anonymous customers place and receive orders securely with privacy safeguards.

Use Case 9: Staff Generates Sales and Order Reports

Preconditions

Staff has reporting permissions, and order history data exists.

Main Flow

1. Staff opens "Reports and Analytics."
2. Selects type (daily sales, popular items, trends).
3. Defines filters (date, category, customer type).
4. System queries database and generates formatted report.
5. Staff reviews results and exports as needed.

Subflows & Alternatives

- Insufficient data → system suggests alternate range.
- Report generation fails → queued for later.
- Export error → alternative formats provided.

Outcome

Reports give insights into sales and customer behavior, aiding decisions.

Use Case 10: System Handles Emergency Incident Response

Preconditions

Emergency protocols are configured, incident interface is accessible, and customer contact channels are active.

Main Flow

1. Incident (e.g., food safety, outage) is identified.
2. Administrator opens "Incident Management" and selects type.
3. System identifies affected customers and orders.
4. Admin disables affected menu items or functions.
5. Notifications are sent to customers with instructions.
6. Refunds are issued, and incident log is generated.

Subflows & Alternatives

- Ingredient tracing identifies specific affected orders.
- Notification failure → backup method used.
- Refund delays logged for manual processing.

Outcome

Incidents are contained quickly with customers informed, refunds handled, and reports generated for compliance.

Summary: Final Analysis - LLM Response Patterns to Prompting

Our comparative analysis revealed distinct response patterns that illuminate how different LLMs interpret and execute the same analytical task. **Perplexity demonstrated the most expansive interpretive approach**, treating prompts as launching points for comprehensive exploration, often surfacing unexpected stakeholders like environmental groups and fraud detection scenarios that, while valuable for brainstorming, sometimes exceeded immediate project scope. **Gemini showed the strongest domain-specific activation**, responding to business analysis prompts by automatically incorporating deep regulatory knowledge and real-world compliance considerations, suggesting more specialized training or fine-tuning for professional contexts. **The WolfCafe-focused approach exhibited superior constraint adherence**, maintaining tight alignment with provided specifications while sacrificing some creative breadth, indicating either more conservative training or better instruction-following optimization. **The systematic methodology approach revealed the most meta-cognitive awareness**, not just answering the prompt but analyzing the process of answering, providing frameworks for stakeholder identification that transcended the immediate task.

This comprehensive analysis provides the foundation for understanding the WolfCafe system's stakeholders, their potential conflicts, and key system functionality through detailed use cases. The systematic approach to stakeholder identification and the use of careful prompting ensures a thorough analysis of the project's requirements, demonstrating sophisticated analytical thinking appropriate for software engineering coursework.