CSC510 Fall 2025: Software Engineering

Proj1c1 Solutions

Group number: 12

Github Repository Link:
https://github.com/Asoingbob225/CSC510

1. Use Cases

1.1 Use Case 01 — Guest Checkout (Friction-Light Ordering)

This use case describes how a user can complete an order without needing to log in, providing only essential details for pickup and payment to reduce friction.

Preconditions

- The menu is visible.
- The user's cart contains at least one item.

Main Flow

A user selects the "Checkout as Guest" option. They are then prompted to provide a name for pickup and enter their payment details. Upon successful payment, the system confirms the order and provides a unique confirmation ID. Key steps include [Minimal contact capture] and [Card tokenization]. Failure to pay triggers [Payment declined].

Subflows

- [Minimal contact capture]: System collects only the necessary information for order pickup.
- [Card tokenization]: System securely processes card information for payment.

Alternative Flows

- [Payment declined]: If payment fails, the system prompts the user to retry or select a different payment method.

1.2 Use Case 02 — One-Tap Modifiers (Preset Customizations)

This use case describes how a user can quickly customize their order by selecting from predefined options, which instantly updates the cart.

Preconditions

- Menu items have predefined, safe presets configured (e.g., "extra shot," "light ice").

Main Flow

A user taps on preset modifier chips displayed with a menu item. The selection is instantly applied, and the cart's contents and total price are updated immediately. The system enforces [Caps per item] to manage preparation complexity. Any conflicts are handled by [Preset conflicts].

Subflows

- [Caps per item]: System limits the number of modifiers that can be applied to a single item to prevent overly complex orders.

Alternative Flows

- [Preset conflicts]: If conflicting presets are selected, the system defaults to the most recent selection and updates the order state.

1.3 Use Case 03 — Live Prep Queue Board (Low-Latency Staff View)

This use case describes how staff members view, claim, and manage incoming orders from a live, first-in-first-out queue.

Preconditions

- There are new orders with the status "Pending".

Main Flow

A staff member views a FIFO queue of pending orders. They can [Claim] an order, which locks it to prevent others from working on it. After preparing the item, they mark it as [Complete], which changes the order status to "Ready" and triggers an inventory adjustment. The system may require staff to handle [Need clarification] or [Low stock] situations.

Subflows

- [Claim]: Staff member selects an order from the queue, locking it to their account.
- [Complete]: Staff member marks the order as prepared, triggering a status change and customer notification.

Alternative Flows

- [Need clarification]: If an order is unclear, staff can use Use Case 06 to contact the customer.
- [Low stock]: If an ingredient is low, staff can use Use Case 04 to manage substitutions.

1.4 Use Case 04 — Smart Substitution Prompt (Stock-Aware)

This use case describes how the system assists staff in handling out-of-stock ingredients by suggesting substitutions and managing customer approval.

Preconditions

- The system has detected a low stock alert for an ingredient during order preparation (from UC2).

Main Flow

When an ingredient for an order is out of stock, the system suggests the nearest available substitute. A staff member sends this suggestion to the customer for approval using Use Case 06. If the customer consents, the order's recipe and price are updated. If the [Customer declines], the item is removed.

Subflows

- There are no specific subflows for this use case.

Alternative Flows

- [Customer declines]: If the customer rejects the suggested substitution, the item is removed from their order, and they are only charged for the remaining items.

1.5 Use Case 05 — Pickup ETA (Simple Heuristic)

This use case describes how the system provides customers with an estimated time of arrival (ETA) for their order based on simple calculations.

Preconditions

- An order has been claimed and is in the "In-Prep" state.

Main Flow

The system displays an estimated pickup time for an order. This ETA is calculated based on the average preparation time for the items ordered and the current length of the prep queue. The ETA is updated whenever the order's status changes. The ETA is also affected by [Capacity throttling].

Subflows

- There are no specific subflows for this use case.

Alternative Flows

- [Capacity throttling]: If the kitchen is at capacity (Use Case 07), the system automatically adjusts the ETA to reflect the longer wait time.

1.6 Use Case 06 — Templated Order Messages (Staff↔Customer)

This use case describes how staff can communicate with customers about order issues using predefined message templates.

Preconditions

- An order requires clarification or a substitution.

Main Flow

A staff member selects a predefined message template (e.g., to ask for approval for a substitution). The system sends this message to the customer via a push notification or in-app message. The customer can reply with a simple "Yes" or "No," and the order is updated accordingly. If there is [No response], a default policy is applied.

Subflows

- There are no specific subflows for this use case.

Alternative Flows

- [No response]: If the customer does not reply within a set time, the system follows a default procedure, such as removing the item or making a standard substitution.

1.7 Use Case 07 — Capacity Throttling (Queue Guardrails)

This use case describes how the system manages incoming order flow during busy periods by adjusting pickup times or temporarily pausing new orders.

Preconditions

- System is configured with staff counts and order capacity thresholds.

Main Flow

If the number of "in-prep" orders exceeds a configured threshold, the system automatically adjusts the checkout process. It will either show the earliest available future pickup time or temporarily pause the ability to place new orders until the queue clears. The system can [Auto-resume] when the load normalizes, or a [Manager override] can be used.

Subflows

- [Auto-resume]: When the number of in-prep orders falls below the threshold, the system automatically returns to normal operation.

Alternative Flows

- [Manager override]: A manager has the authority to manually disable throttling to accommodate special circumstances.

1.8 Use Case 08 — Accessibility Mode Toggle (WCAG-Oriented)

This use case describes how users can enable an accessibility-focused viewing mode that adjusts the UI to meet WCAG standards.

Preconditions

- The user interface is built to support accessibility presets like reduced motion, high contrast, and text scaling.

Main Flow

A user activates the "Accessibility Mode" via a toggle. The system immediately applies a set of WCAG-aligned UI presets, such as high-contrast colors and reduced animations, and remembers this preference for future sessions. The system can also [Detect OS-level pref] to enable this automatically.

Subflows

- [Detect OS-level pref]: On the user's first visit, the system can check for operating system-level accessibility settings and automatically enable the mode.

Alternative Flows

- There are no specific alternative flows for this use case.

1.9 Use Case 09 — “86 Item Now” (Instant Availability Toggle)

This use case describes how staff can instantly mark a menu item as unavailable (86'd), hiding it from the customer-facing menu.

Preconditions

- Menu items exist in the system.

Main Flow

From the staff-facing queue board, a staff member can tap an "86" button next to an item. This action immediately hides the item from the menu, preventing new orders. The staff member can set an optional auto-restore time. Staff can also [Restore early].

Subflows

- There are no specific subflows for this use case.

Alternative Flows

- [Restore early]: A staff member can manually make the item available again before the auto-restore time has elapsed.

1.10 Use Case 10 — Promo Code MVP (Single-Field Apply)

This use case describes how a customer can apply a promotional code to their order to receive a discount.

Preconditions

- The checkout process includes a hook for applying coupons.

Main Flow

A customer enters a promo code into a single field during checkout. The system validates the code and, if valid, applies the corresponding discount to the order total. If the code is [Expired/invalid], an inline error message is displayed.

Subflows

- There are no specific subflows for this use case.

Alternative Flows

- [Expired/invalid]: If the entered code is not valid, the system shows a message explaining the issue and does not apply a discount.

1.11 Use Case 11 — Basic Sales Snapshot (Today/Week)

This use case describes how an administrator can view a simple dashboard with key sales metrics for the day or week.

Preconditions

- Completed orders exist in the system.

Main Flow

An administrator views a dashboard displaying tiles for total revenue, number of orders, and the average ticket price. They can filter this data by location and time period. The admin can also perform a [CSV export]. If there is [No data], a message is shown.

Subflows

- [CSV export]: The admin can export the aggregated sales data as a CSV file.

Alternative Flows

- [No data]: If there are no transactions for the selected period, the system displays a "No transactions" message.

1.12 Use Case 12 — Customer-Facing Ready Board (Pickup Screen)

This use case describes a large-screen display in the pickup area that shows customers which orders are ready.

Preconditions

- Orders have been marked as "Ready".

Main Flow

A large display screen cycles through a list of orders that are ready for pickup, showing the order number and customer name. A privacy mode is available to show only initials or the order code. The board automatically refreshes every few seconds. If a [Notification failed], the board still serves as a reliable indicator.

Subflows

- There are no specific subflows for this use case.

Alternative Flows

- [Notification failed]: Even if the customer's primary notification (e.g., push notification) fails, their order number will still appear on the board when it is ready.

1.13 Use Case 13 — Favorites & One-Tap Reorder

This use case describes how a logged-in customer can save a past order as a favorite and quickly reorder it.

Preconditions

- The customer has at least one past order in their history.

Main Flow

A customer views their order history and saves a specific order as a "Favorite". From their favorites list, they can use a single tap to re-add all items from that past order into their current cart. The system handles [Unavailable items] gracefully. If the user has [No history], the feature is hidden.

Subflows

- There are no specific subflows for this use case.

Alternative Flows

- [Unavailable items]: If an item from the favorite order is currently unavailable, it is not added to the cart, and the user is notified.
- [No history]: If a customer has no previous orders, the favorites feature is not displayed.

1.14 Use Case 14 — Schedule for Later (Pickup Window)

This use case describes how a customer can place an order in advance and schedule it for a future pickup time.

Preconditions

- The system is configured with rules for order capacity and scheduling.

Main Flow

During checkout, a customer can select a specific time slot for a future pickup. The available time slots are determined by the system's capacity throttling rules (Use Case 07). The order's ETA is updated to reflect the scheduled time.

Subflows

- There are no specific subflows for this use case.

Alternative Flows

- There are no specific alternative flows for this use case.

1.15 Use Case 15 — Saved Dietary Profile

This use case describes how a user can save their dietary preferences in their profile to pre-filter menus.

Preconditions

- The user is authenticated and accessing their profile.

Main Flow

A user saves their dietary preferences (e.g., "vegan," "no nuts") in their profile. When they browse the menu, the system uses this information to automatically filter the displayed items, showing only those that match their profile.

Subflows

- There are no specific subflows for this use case.

Alternative Flows

- There are no specific alternative flows for this use case.

1.16 Use Case 16 — Tip Suggestions A/B

This use case describes how the system can A/B test different tip suggestion options to determine which is most effective.

Preconditions

- The payment checkout flow includes a step for adding a tip.

Main Flow

The system presents different tipping options to different users. For example, one group of users might see percentage-based suggestions (15%, 18%, 20%), while another group sees options to round up to the nearest dollar. The system tracks the results to determine which method generates higher tips.

Subflows

- There are no specific subflows for this use case.

Alternative Flows

- There are no specific alternative flows for this use case.

2. Reflections

2.1 Decisions About What Not to Do

Tax Management (UC11, UC14, UC21-23, UC25): All functionalities related to calculating, configuring, reporting, and auditing sales tax were deferred. The strategic reasoning is that tax compliance is a complex, jurisdiction-dependent problem that is high-effort and low-impact for validating the core order-and-fulfill loop. Solving this adds no immediate value to the primary users (Customers, Staff) and would significantly delay the launch. The business can fall back on a manual process of remitting taxes based on total sales, which is viable at a small scale.

Payroll and HR Management (UC24, UC26, UC34): Use cases for tracking employee wages, tips, and tax withholdings were excluded. This decision stems from the understanding that payroll is a solved problem best handled by specialized third-party systems. Building these features internally would be redundant, expensive, and distract the development team from the project's unique challenges.

Complex Financial and Programmatic Integrations (UC16, UC29): Support for nutritional benefit programs like WIC or SNAP was deemed out of scope. These integrations require certification and complex validation logic that would divert significant resources from the core product.

2.2 Potential Negative Impacts or Disappointments

This focused MVP strategy, while effective for rapid validation, inevitably disappoints stakeholders whose primary interests lie outside the core user journey. By prioritizing the `Customer -> Staff -> Customer` loop, we have consciously accepted the risk of creating dissatisfaction among the following groups:

- **Administrators/Managers:** This group experiences the most significant feature loss. They lose the detailed operational tools they need for efficient oversight, such as comprehensive sales reporting, full menu and inventory management, and staff performance tracking. The MVP provides only a basic sales snapshot and a simple item availability toggle, leaving them without the granular control and data analytics needed for optimizing revenue and managing daily operations effectively.
- **Business Owners & University Dining Services:** These stakeholders are focused on profitability, compliance, and reputation. They will be disappointed by the exclusion of features related to long-term customer adoption, such as mandatory logins for loyalty programs (the MVP explicitly prioritizes guest checkout to reduce friction). More critically, the complete deferral of all tax, food safety, and regulatory compliance features (UC11-UC34) creates significant business and legal risk, directly conflicting with their core objectives of maintaining a good reputation and ensuring compliance with university and legal standards.

- **Regulatory Bodies:** This stakeholder group is entirely unserved by the MVP. The absence of features for allergen management (UC15), health inspections (UC12), or data security means the product, in its initial form, would not meet legal standards for public operation. While this is a reasonable simplification for an academic project, it represents the largest gap between the MVP and a production-ready system.

2.3 Changes Made to the MVP for Appeasing Stakeholders

Prioritizing the Core User Experience
The most significant modification was the strategic decision to prioritize the essential `Customer → Staff → Customer` workflow. By deferring complex administrative and regulatory features, the team focused its resources on perfecting the core loop of ordering, preparation, and notification. This "move fast" approach was intended to satisfy the most critical stakeholders: **Customers**, who value fast service and convenience, and **WolfCafe Staff**, who need a simple, manageable workflow to function effectively under pressure. This focus ensures that the MVP delivers immediate, tangible value to its primary users.

Streamlining Features for Usability and Speed
To support the core loop, many original use cases were simplified to make them faster and easier to use. This directly addresses known tensions between usability and feature complexity. For example, instead of forcing profile creation (UC4), the MVP prioritizes a Guest Checkout (MVP UC01). This resolves the documented conflict between
Customers, who prefer minimal logins, and Administrators, who want to track users for loyalty programs. Similarly, rather than implementing a comprehensive inventory management system (UC6), the MVP provides a simple, one-tap '86 Item Now' button (MVP UC09). This gives WolfCafe Staff a tactical tool that is fast and simple, which is what they need during a busy shift.

Resolving Stakeholder Conflicts
Where stakeholder needs were in direct opposition, the team implemented features to address the conflict. The stakeholder analysis identified a clash between the Dev Team, who may favor a "flashy" UI, and Accessibility Users, who require a readable and compliant interface. To address this, an Accessibility Mode Toggle (MVP UC08) was included in the MVP. This feature allows the Dev Team to build a modern default interface while enabling Accessibility Users to switch to a WCAG-aligned mode. This approach treats accessibility as a distinct feature and addresses the needs of both stakeholder groups.

3. Prompt History

3.1 ChatGPT5 Thinking

https://chatgpt.com/share/e/68c48e91-dc20-8000-871e-bf676ef583b1

3.2 Gemini 2.5 Pro

https://g.co/gemini/share/74ab1a3a964a