Use Cases

UC1: User Creates an Account

- **Preconditions**
  - The user does not have an existing account.
  - The user has provided a valid, unique email address.
- **Main Flow**
  - The user navigates to the application's "Sign Up" or "Create Account" screen.
  - The user enters their first name, last name, a valid email address, and a password into the respective fields.
  - The user confirms their password by re-entering it.
  - The user clicks the "Create Account" button.
  - The system validates that the email is not already in use and that the passwords match.
  - The system creates a new customer record in the database.
  - The system automatically logs the user into their new account and redirects them to the main menu screen.
- **Subflows**
  - **[Password Strength Indicator]** As the user types their password, the system displays a real-time indicator of its strength (e.g., Weak, Medium, Strong).
- **Alternative Flows**
  - **[Email Already Exists]** If the provided email address is already associated with an existing account, the system will display an error message: "This email is already in use. Please log in or use a different email."
  - **[Passwords Do Not Match]** If the password and password confirmation fields do not match, the system will prevent submission and display an error message: "Passwords do not match."
  - **[Invalid Email Format]** If the email address is not in a valid format (e.g., missing '@'), the system will display an error message requesting a valid email.

UC2: User Logs In

- **Preconditions**
  - The user has an existing account with a known email and password.
- **Main Flow**
  - The user navigates to the "Log In" screen.
  - The user enters their registered email address and password.
  - The user clicks the "Log In" button.
  - The system verifies the credentials against the user database.
  - Upon successful authentication, the system creates a user session and directs the customer to the main menu screen.

- **Subflows**
    - **[Navigate to Password Reset]** If the user has forgotten their password, they can click a "Forgot Password?" link, which takes them to the password reset flow (UC33).
- **Alternative Flows**
    - **[Invalid Credentials]** If the email and password combination does not match a record in the database, the system will display a generic error message: "Invalid email or password." The system will not specify which of the two was incorrect.
    - **[Account Locked]** If the account has been locked due to too many failed login attempts, the system will display a message: "Your account has been temporarily locked for security reasons. Please try again later or reset your password."

UC3: User Resets a Forgotten Password

- **Preconditions**
    - The user has an existing account but has forgotten their password.
    - The user has access to the email account they registered with.
- **Main Flow**
    - The user clicks the "Forgot Password?" link on the login screen.
    - The system prompts the user to enter the email address associated with their account.
    - The user enters their email and submits the form.
    - The system verifies that the email exists in the database.
    - The system generates a secure, single-use password reset link and sends it to the user's email address.
    - The system displays a confirmation message on the screen: "If an account exists for this email, a password reset link has been sent."
    - The user opens their email, clicks the link, and is taken to a "Create New Password" screen.
    - The user enters and confirms their new password.
    - The system updates the user's account with the new password and logs them in.
- **Subflows**
    - N/A
- **Alternative Flows**
    - **[Email Not Found]** If the entered email address does not exist in the database, the system will still display the same generic confirmation message ("...a password reset link has been sent") to prevent account enumeration, where a malicious actor could check for valid emails.
    - **[Reset Link Expired]** If the user clicks an old or expired reset link, the system will display a message: "This password reset link is invalid or has expired. Please request a new one."

UC4: User Views the Menu

- **Preconditions**
  - The user has opened the application.
  - The administrator has created at least one menu category and added at least one menu item.
- **Main Flow**
  - After logging in or launching the app, the user is presented with the main menu view.
  - The system displays a list of menu categories (e.g., "Coffee," "Pastries," "Sandwiches").
  - The user selects a category.
  - The system displays all available menu items within that category.
  - For each item, the system shows its name, a brief description, and its price.
  - The user can scroll through the items and select other categories to browse the entire menu.
- **Subflows**
  - **[View Item Details]** The user can tap on a specific menu item to navigate to a dedicated detail page, which may show a larger image and a more detailed description.
- **Alternative Flows**
  - **[Item Unavailable]** If a menu item is marked as "Out of Stock" by the staff, it will be grayed out or display an "Unavailable" banner, and the user will not be able to add it to their cart.
  - **[Empty Menu]** If no menu items have been configured by the administrator, the system will display a message like: "Our menu is currently being updated. Please check back soon!"

UC5: User Adds/Removes Item from Cart

- **Preconditions**
  - The user is viewing the menu (UC34).
- **Main Flow**
  - The user finds an item they wish to purchase.
  - The user clicks an "Add to Cart" or "+" button associated with the item.
  - The system adds one unit of the selected item to the user's shopping cart.
  - The system provides visual feedback, such as updating a cart icon with the total number of items.
  - The user navigates to the cart screen by tapping the cart icon.
  - The user locates an item in the cart they wish to remove and clicks a "Remove" or "X" button next to it.
  - The system removes the item from the cart and updates the order subtotal.
- **Subflows**

    - **[Continue Shopping]** After adding an item, the user can remain on the menu to continue browsing and adding more items.
- **Alternative Flows**
    - **[Add Unavailable Item]** The system will prevent the user from adding an item that is marked as "Unavailable," and the "Add to Cart" button will be disabled for that item.
    - **[Removing Last Item]** If the user removes the last item from their cart, the cart will display an "Your cart is empty" message.

UC6: User Modifies Item Quantity in Cart

- **Preconditions**
    - The user has at least one item in their shopping cart.
    - The user is on the cart screen.
- **Main Flow**
    - The user navigates to their shopping cart.
    - Next to an item in the cart list, the system displays its current quantity along with "+" and "-" controls.
    - The user clicks the "+" button to increase the quantity of the item by one.
    - The system updates the line-item total and the order subtotal.
    - The user clicks the "-" button to decrease the quantity of the item by one.
    - The system updates the line-item total and the order subtotal.
- **Subflows**
    - N/A
- **Alternative Flows**
    - **[Decrease Quantity to Zero]** If the user decreases an item's quantity to zero (e.g., by pressing "-" when the quantity is 1), the system will automatically remove the item from the cart entirely.
    - **[Inventory Limit Reached]** For an MVP, inventory is likely managed at a simple "available/unavailable" level. In a more advanced version, if a user tried to increase the quantity beyond the available stock, the system would prevent it and show a "Maximum quantity reached" message.

UC7: User Enters Payment Information

- **Preconditions**
    - The user has at least one item in their cart.
    - The user has proceeded to the "Checkout" or "Payment" screen.
- **Main Flow**
    - On the payment screen, the system displays the final order total, including taxes.
    - The system presents a secure form for credit card information.

    - The user enters their credit card number, expiration date, CVC code, and the billing zip code.
    - As the user types, the system provides real-time validation for field formatting (e.g., ensuring the card number has the correct number of digits).
    - Once all fields are correctly filled, the "Place Order" button becomes active.
    - The user clicks "Place Order" to submit the payment for processing.
- **Subflows**
    - N/A
- **Alternative Flows**
    - **[Invalid Card Details]** If the payment processor declines the transaction due to invalid card information (e.g., incorrect number, expired card), the system will display an error message: "Payment failed. Please check your card details and try again." The form fields will be preserved for easy correction.
    - **[Insufficient Funds]** If the transaction is declined due to insufficient funds, the system will display a generic failure message to protect user privacy, such as: "Your payment could not be processed. Please try a different card."

UC8: Staff Views a Specific Order's Details

- **Preconditions**
    - A staff member is logged into the system with "Order Fulfillment" permissions.
    - At least one customer order has been submitted and is in the "Pending" queue.
- **Main Flow**
    - The staff member navigates to the main order queue screen, which displays a list of incoming orders.
    - Each order in the list shows summary information: the order number, customer name, and time of placement.
    - The staff member selects an order from the list to begin working on it.
    - The system navigates to a detailed view for that specific order.
    - This view clearly lists all items in the order, including the quantity of each.
    - The system marks the order as "In Progress," which removes it from the main "Pending" queue to prevent other staff from working on the same order.
- **Subflows**
    - **[Print Order Ticket]** The staff member can press a "Print" button to generate a physical ticket for the kitchen staff, containing all the order details.
- **Alternative Flows**
    - **[Order Canceled]** If the customer cancels the order while it is still in the "Pending" queue, the order will be removed from the list before a staff member can select it.

UC9: Staff Marks an Order as 'Completed'

- **Preconditions**
  - A staff member is logged in.
  - An order has already been prepared and marked as "Ready for Pickup."
  - The customer has arrived and collected their order.
- **Main Flow**
  - The staff member navigates to the "Ready for Pickup" queue in the staff interface.
  - The staff member identifies the customer's order in the list (e.g., by asking for the customer's name or order number).
  - After handing the order to the customer, the staff member taps a "Mark as Completed" button next to the order.
  - The system moves the order from the active "Ready for Pickup" queue to a "Completed Orders" historical log.
  - This action signifies the end of the order lifecycle from the staff's perspective.
- **Subflows**
  - N/A
- **Alternative Flows**
  - **[Mark Completed by Mistake]** If a staff member marks an order as completed by accident, a user with "Manager" permissions can access the historical log and revert the order's status back to "Ready for Pickup."

UC10: Admin Views a Simple List of Daily Orders

- **Preconditions**
  - An administrator or manager is logged into the system.
- **Main Flow**
  - The administrator navigates to the "Reports" or "Sales History" section of the admin dashboard.
  - The system defaults to showing today's date.
  - The system displays a simple, chronological list of all orders that have been placed on that date.
  - For each order, the list shows the order number, the time it was placed, the customer's name, and the final order total.
  - At the bottom of the list, the system displays the total sales revenue for the selected day.
- **Subflows**
  - **[Change Date]** The administrator can use a date picker to select a different day in the past to view its order list and sales total.
- **Alternative Flows**
  - **[No Orders]** If no orders were placed on the selected date, the system will display a message: "No sales data for this day."

Reflection

How did you decide what NOT to do?

To decide what not to do, we focused on the definition of an MVP: the simplest version of a new product with just enough core features to attract early-adopter customers and validate a business hypothesis with the least effort and resources. Every feature was evaluated according to this criteria, and if it wasn’t deemed absolutely necessary, then it was removed.

This led to the removal of three main categories of features:

1. **Growth and Optimization Features:** This category includes loyalty programs, discount codes, and detailed reporting dashboards. While beneficial for long-term business growth and marketing, these tools are designed to optimize and scale an existing user base. However, the MVP's primary goal is to first validate that a user base *can be established at all*. Thus, this was deferred to later possible implementations of the product.
2. **Convenience and Power-User Features:** This group contains functions like re-ordering from order history, saving payment methods, and setting a favorite order. These features dramatically improve the experience for *repeat customers*. However, they are not necessary for the initial transaction and add significant development complexity. The MVP is focused on validating the value proposition for a new user, not perfecting the journey for a loyal one.
3. **High-Effort Compliance and Data Features:** Features such as detailed nutritional/allergen management and automated data privacy requests (account deletion, data downloads) were deferred. While critically important for a mature, at-scale product, they represent a disproportionately high development effort for an initial launch. In the early stages, manual processes can fulfill the few data requests that might arise, and a simple disclaimer can address allergen concerns until a robust system is built.

By ruthlessly prioritizing the core transaction, we ensured the MVP remained lean and focused.

What negative impacts or disappointments this MVP could have for your stakeholders.

Our minimalist MVP, while strategic, may disappoint stakeholders in the following ways:

- **Customers:**
  - **High Friction:** Forced account registration and the lack of saved payment methods will feel inconvenient compared to competing apps.
  - **Missing Information:** The absence of nutritional and allergen data will make the app unusable for customers with dietary restrictions.
  - **No Incentives:** Lack of a loyalty program or discounts gives power users no reason to engage beyond a single transaction.

- **Cafe Owner & Management:**
    - **Limited Business Insight:** A simple sales list offers no way to identify trends or track item performance, hindering data-driven decisions.
    - **Inability to Market:** Without discount codes, management cannot run promotions to attract new users.
- **Cafe Staff:**
    - **Workflow Rigidity:** The inability to reverse an order's status (e.g., from "Ready" back to "In Progress") could complicate handling simple mistakes.

What changes you made (and why) to the MVP to appease at least some of the stakeholders.

Recognizing the risks of a purely austere MVP, we added three high-impact features to improve adoption and stakeholder satisfaction:

1. **Guest Checkout:** We introduced a guest checkout flow to reduce friction for new users. This allows customers to place an order with only an email, directly addressing the most significant barrier to a first-time purchase.
2. **"Daily Pulse" Admin Dashboard:** The admin view was upgraded to a dashboard displaying total revenue, order volume, and the top 5 best-selling items. This gives the owner immediate, actionable insight into daily performance without the complexity of a full reporting suite.
3. **Reversible Order Status for Staff:** We added a button allowing staff to move an order from "Ready for Pickup" back to "In Progress." This simple change empowers staff to easily correct common errors, making the workflow more flexible and efficient.

Prompt History

I couldn’t find a way to share a public link to the Gemini chats I used that were accessible to other accounts other than the one I used to prompt the LLM. Instead, here are the prompts that I used.

Prompt 1

Your goal is a MVP (minimal viable product): the most basic version of a product that allows a team to collect the maximum amount of validated learning about customers with the least amount of effort. Decide what from 1b1 you will NOT be doing, and justify why you will not be doing that.

Included files: Project1b1.pdf

Prompt 2

Generate at least 10 use cases (5 pages total) describing the new design of the minimal viable product, each with:

- Preconditions
- Main Flow
- Subflows
- Alternative Flows

Prompt 3

Reflection document (max 2 pages): How did you decide what NOT to do? What negative impacts or disappointments this MVP could have for your stakeholders. What changes you made (and why) to the MVP to appease at least some of the stakeholders.

Prompt 4

Make these two sections more concise:

What negative impacts or disappointments this MVP could have for your stakeholders.

What changes you made (and why) to the MVP to appease at least some of the stakeholders.