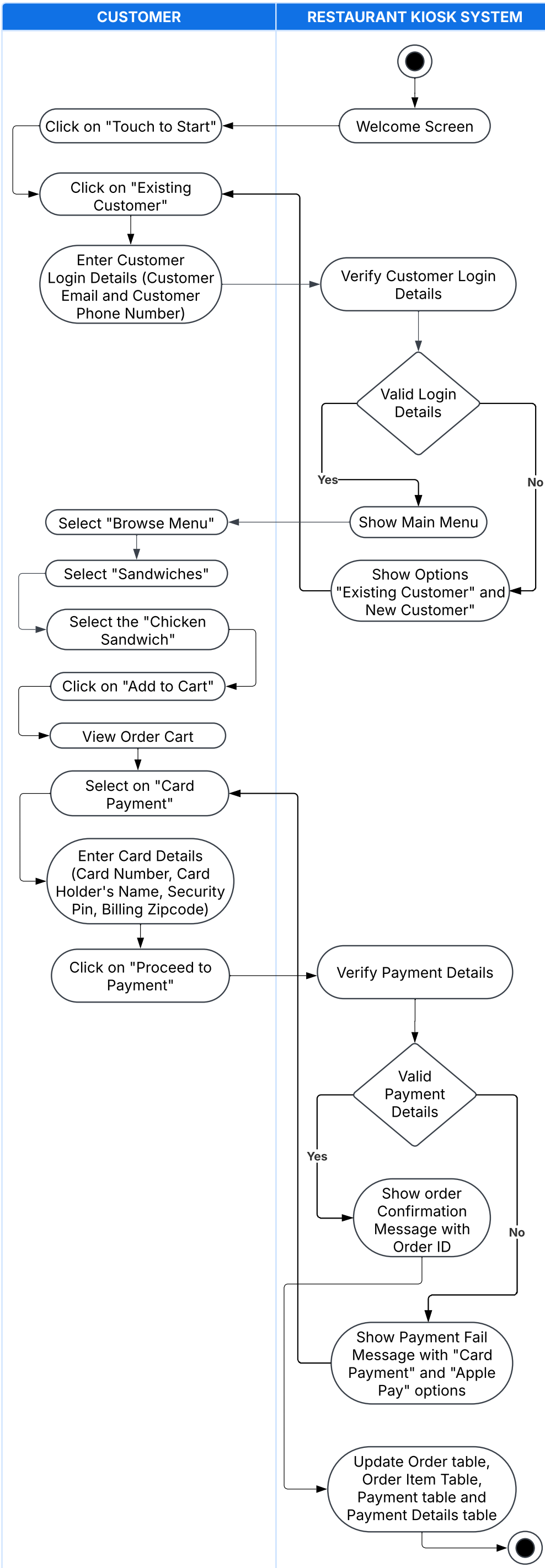
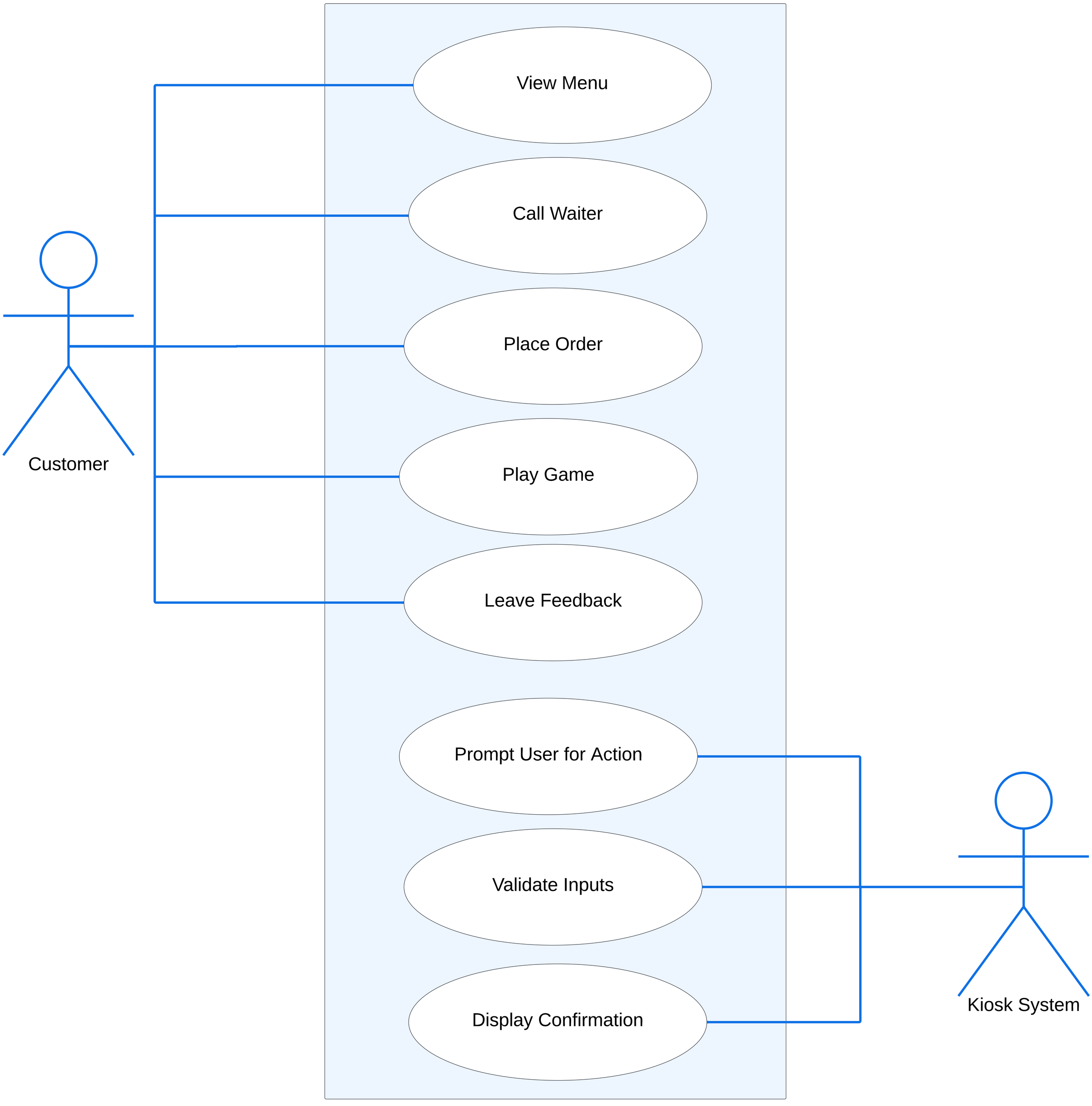
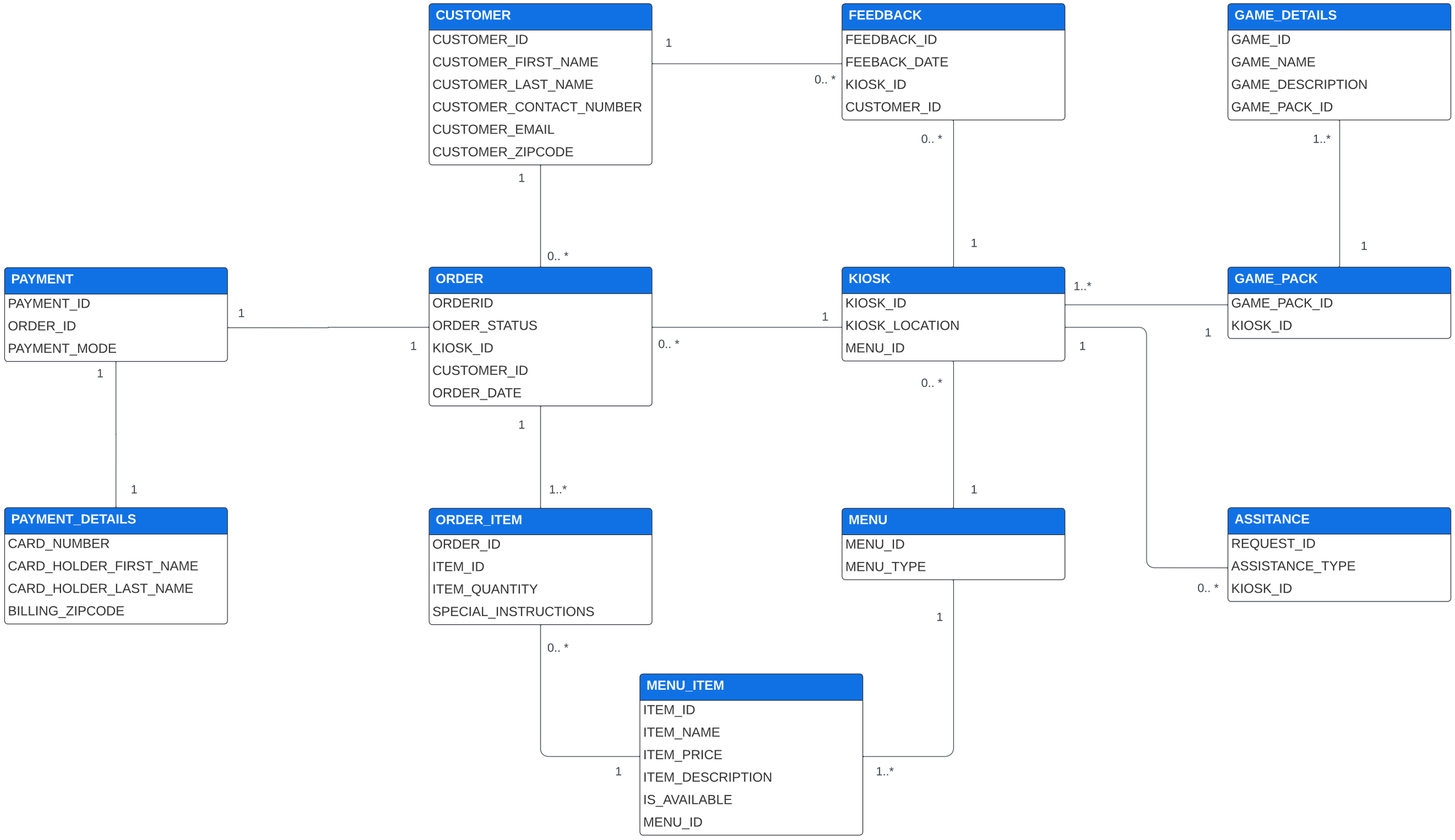
ACTIVITY DIAGRAM



Use Case Name	Place Order and Make Payment	
Scenario	An existing customer logs into the kiosk, selects food items, and completes payment.	
Triggering Event	Customer initiates interaction by touching the kiosk interface.	
Brief Description	The customer initiates the order process at the kiosk by logging in using email and phone number. After successful authentication, the customer browses the categorized menu, selects items, adds them to the cart, and proceeds to payment. The system verifies card details and processes the payment. Upon success, the order is confirmed, stored in the system, and a confirmation message with order ID is displayed. If the payment fails, the system prompts the customer to try a different payment method.	
Actors	Customer, Restaurant Kiosk System	
Related Use Cases	• View Menu • Authenticate Customer • Process Payment • Display Confirmation • Handle Payment Exception	
Stakeholders	Customer, Kitchen Staff, Restaurant Management, IT Support Team	
Preconditions	• Kiosk is powered on and connected to the backend • Menu is up to date • Payment gateway is active • Customer account with valid credentials exists in the system	
Postconditions	• Order and payment details are stored in respective tables • Order is sent to kitchen staff for preparation • Customer receives confirmation and order ID • Payment failure is logged if applicable	
Flow of Activities	**Actor** 1. Clicks on “Touch to Start” 2. Selects “Existing Customer” 3. Enters login details (email and phone number) 4. Selects “Browse Menu” 5. Navigates to “Sandwiches” → selects “Chicken Sandwich” 6. Clicks “Add to Cart” and reviews order 7. Selects “Card Payment” 8. Enters card details (card number, name, security pin, billing zip) 9. Clicks “Proceed to Payment”	**System** • Displays welcome screen and login options • Verifies login credentials • If invalid → return to login • If valid → display main menu • Shows categorized menu • Displays cart and payment options • Verifies payment details • If valid → show confirmation with order ID • If invalid → show failure message with retry options • Updates order, order item, payment, and payment detail tables
Exception Conditions	• Invalid login → redirect to login options • Outdated menu → error message, refresh attempt • Invalid card details → retry prompt with other payment methods • System timeout or network error during payment → prompt to re-enter or switch method • Backend failure in updating tables → log error and display fallback message	

Restaurant Kiosk System




CUSTOMER
CUSTOMER_ID
CUSTOMER_FIRST_NAME
CUSTOMER_LAST_NAME
CUSTOMER_CONTACT_NUMBER
CUSTOMER_EMAIL
CUSTOMER_ZIPCODE
FEEDBACK
FEEDBACK_ID
FEEBACK_DATE
KIOSK_ID
CUSTOMER_ID
GAME_DETAILS
GAME_ID
GAME_NAME
GAME_DESCRIPTION
GAME_PACK_ID
PAYMENT
PAYMENT_ID
ORDER_ID
PAYMENT_MODE
ORDER
ORDERID
ORDER_STATUS
KIOSK_ID
CUSTOMER_ID
ORDER_DATE
KIOSK
KIOSK_ID
KIOSK_LOCATION
MENU_ID
GAME_PACK
GAME_PACK_ID
KIOSK_ID
PAYMENT_DETAILS
CARD_NUMBER
CARD_HOLDER_FIRST_NAME
CARD_HOLDER_LAST_NAME
BILLING_ZIPCODE
ORDER_ITEM
ORDER_ID
ITEM_ID
ITEM_QUANTITY
SPECIAL_INSTRUCTIONS
MENU
MENU_ID
MENU_TYPE
ASSITANCE
REQUEST_ID
ASSISTANCE_TYPE
KIOSK_ID
MENU_ITEM
ITEM_ID
ITEM_NAME
ITEM_PRICE
ITEM_DESCRIPTION
IS_AVAILABLE
MENU_ID
1
0.. *
1..*

Restaurant Self-Service Kiosk System

Problem Description:

In today's world restaurants are facing a lot of pressure due to labour shortages, rising operational costs, and the customer have expectations for faster, error free service. But with the limited staff available during peak hours, delays in taking orders, mixing up table orders, and high labour costs have all become common issues that restaurants and customers face. Traditional systems also rely heavily on manual intervention, which definitely increases the chances of human errors in order processing and fulfilment.

To address these issues, we are proposing a self-service kiosk system. This kiosk will allow customers to view the menu, call a waiter, request the bill, access entertainment, and provide feedback which can all be done with minimal staff dependency. This system is focused in such a way that it can streamline the customer ordering process, also reduce dependence on staff, and improve operational efficiency.

System Capabilities:

The proposed kiosk will offer intuitive and user-friendly functionality specifically for customers. These capabilities are designed to automate common service tasks, minimize human error, and improve customer engagement.

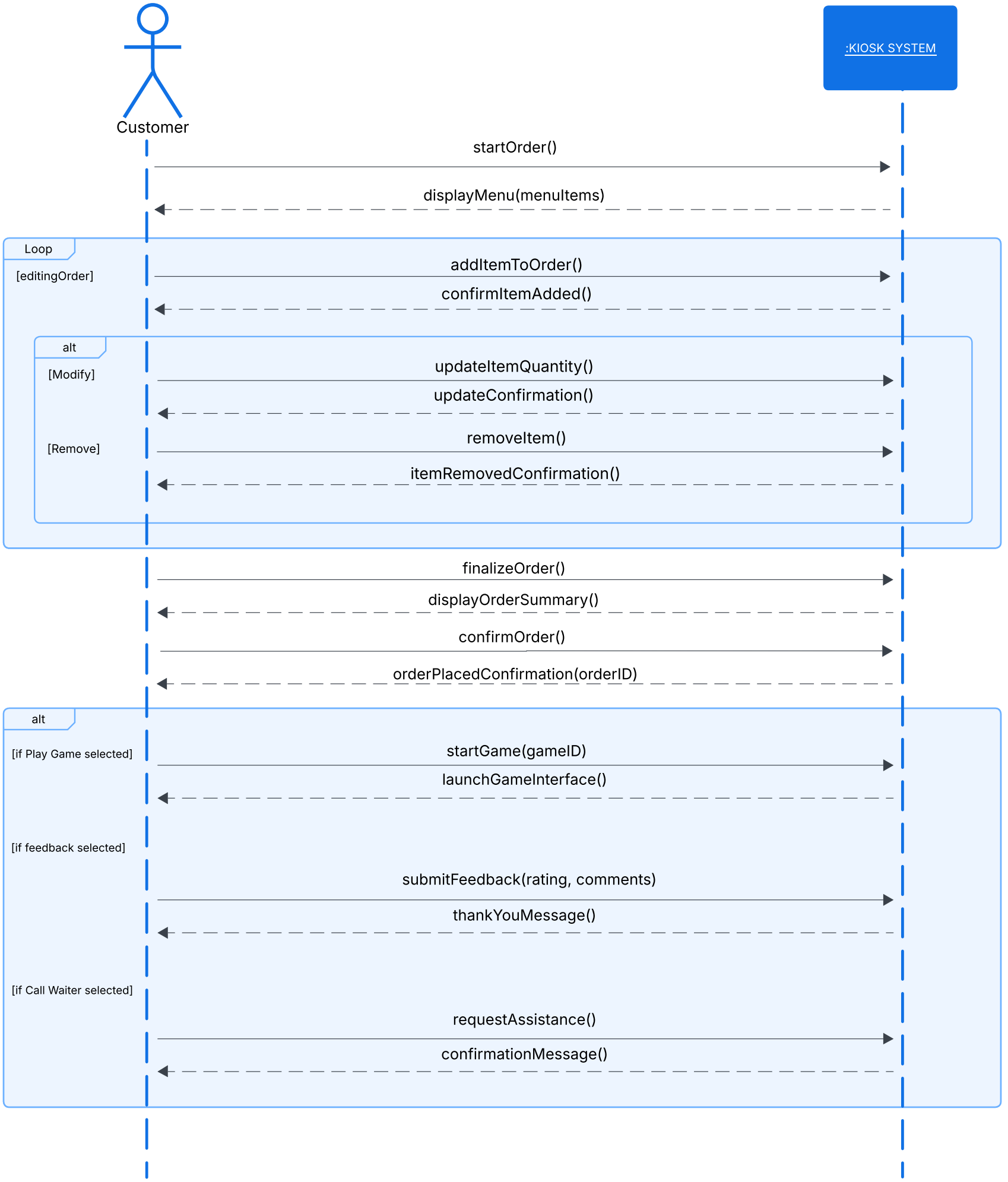
- Display the restaurant menu for browsing
- Allow customers to call a waiter for assistance
- Enable customers to request their bill
- Offer entertainment features (games, media) during waiting periods
- Collect digital customer feedback
- Store all interactions for backend reporting and analysis

Business Benefits:

The implementation of this kiosk system is expected to provide measurable improvements in both operations and customer satisfaction:

- Reduce staff workload by automating frequent customer requests
- Improve customer satisfaction with faster response times

- Increase efficiency and table turnover by accelerating billing and ordering processes
- Enhance brand perception with interactive, modern service
- Capture real-time feedback for quality improvement
- Enable data-driven decisions based on usage and interaction logs


Customer
:KIOSK SYSTEM
startOrder()
displayMenu(menuItems)
Loop
[editingOrder]
addItemToOrder()
confirmItemAdded()
alt
[Modify]
updateItemQuantity()
updateConfirmation()
[Remove]
removeItem()
itemRemovedConfirmation()
finalizeOrder()
displayOrderSummary()
confirmOrder()
orderPlacedConfirmation(orderID)
alt
[if Play Game selected]
startGame(gameID)
launchGameInterface()
[if feedback selected]
submitFeedback(rating, comments)
thankYouMessage()
[if Call Waiter selected]
requestAssistance()
confirmationMessage()