Mehul Agarwal :2023101049
Paarth Jindal :2023101050
Varun Gupta :2023101108
Vanshika Ahalawat :2023113003

Arya Mahendraker:2023101041

1. Use Case: Place Food Order (Fully Dressed)

- **Use Case Number**: OFO-US0001
- **Use Case Name**: Place Food Order
- **Overview**: A customer places an order for food from an online platform (website or mobile app). The system processes the order, calculates the total cost, applies any discounts, handles payment, and notifies the restaurant for preparation.
- **Primary Actor**: Customer

Stakeholders and Interests

- **Customer**: Wants a smooth ordering experience, accurate pricing, estimated delivery time, and secure payment processing.
- **Restaurant Staff**: Wants order details to be correctly transmitted and updated to start preparation immediately.
- **Delivery Person**: Needs order details, delivery location, and payment information for accurate delivery.
- **Company (Platform Owner)**: Wants to accurately log transactions, process payments, and manage order data efficiently.
- **Payment Gateway Provider**: Wants to process payments securely and ensure compliance with financial regulations.
- **Customer Support Team**: Needs order logs and payment details to resolve disputes or refunds.
- **Government Tax Agencies**: Require tax to be calculated and recorded for compliance.

Preconditions

- The customer is registered and logged into the system.

- The restaurant is active and accepting orders.

Postconditions

- Order is successfully placed.
- Payment is processed (or marked for cash on delivery).
- The restaurant receives the order details.
- The estimated delivery time is shown to the customer.

Main Flow

1. The customer browses the menu and selects items to order.
2. The customer adds selected items to the cart.
3. The system updates the cart and calculates the total price, including applicable taxes and discounts.
4. The customer proceeds to checkout.
5. The system prompts the customer to select a delivery address or choose pickup.
6. The customer selects a payment method (Credit Card, UPI, Wallet, or Cash on Delivery).
7. The system processes the payment and confirms the transaction.
8. The order is sent to the restaurant for preparation.
9. The system notifies the restaurant and updates order status to “Processing.”
10. The system displays an estimated delivery time and tracking information.
11. The customer receives an order confirmation via email/SMS.

Alternate Flows

a. Customer modifies order before payment

1. The customer removes or adds items before checkout.
2. The system updates the total price and cart details.

b. Customer cancels order before restaurant confirmation

1. The customer selects "Cancel Order" before the restaurant starts preparation.
2. If paid, the system processes the refund (based on the cancellation policy).
3. The system updates the order status to "Canceled."

c. Payment fails

1. The system notifies the customer about the failed transaction.
2. The customer selects an alternative payment method.
3. The system retries payment processing.
4. If multiple failures occur, customer support intervention may be required.

d. Order is not accepted by the restaurant

1. The restaurant rejects the order due to unavailability of items.
2. The system notifies the customer and offers a refund or alternative choices.

e. Customer requests an urgent delivery

1. The customer selects "Priority Delivery" for an additional charge.
2. The system recalculates the total amount.
3. The delivery personnel are notified about priority handling.

f. Food is unavailable after payment

1. The restaurant informs the system about unavailability.
2. The system suggests alternative dishes or offers a full refund.

Special Requirements

- The platform must process payments within **10 seconds** for 90% of transactions.
- The order tracking system must update every **2 minutes** until delivery.
- The system should allow guest checkouts but require phone/email verification.
- Multi-language support must be available based on the user's location.
- Secure storage of customer payment details (PCI DSS compliance).
- Order history must be retrievable for at least **6 months** for dispute resolution.

Technology & Data Variations

- **Order placement methods**: Web-based, Mobile app, or Voice assistant.
- **Payment options**: Credit/Debit cards, UPI, Digital Wallets, Cash on Delivery.
- **Delivery tracking**: GPS-based real-time tracking via mobile app.
- **Receipt options**: E-receipt via email/SMS, printable receipt on the order page.
- **Discount codes**: System applies valid promo codes at checkout.

2. Use Case: ATM Cash Withdrawal

Unumber: ATM-US0001
Use Case Name: ATM Cash Withdrawal
Overview:
A customer withdraws cash from an ATM by providing authentication and selecting the withdrawal amount. The ATM verifies the account balance, dispenses cash, and updates the account.

Primary Actor: Customer

Stakeholders and Interests:

- **Customer:** Wants a quick, secure, and hassle-free transaction. Needs accurate cash withdrawal and an updated balance.
- **Bank:** Wants to ensure accurate account updates, security of transactions, and fraud prevention.
- **ATM System:** Wants to ensure availability of cash and correct functioning of the machine.
- **Network Payment Processor:** Wants correct digital authorization for the transaction.
- **Government/Regulatory Authorities:** Want compliance with financial transaction regulations and fraud monitoring.

Preconditions:

- ATM is operational and connected to the banking network.
- Customer has an active bank account with sufficient funds.
- Customers have a valid ATM/debit card or support cardless withdrawals.

Postconditions:

- Cash is dispensed.
- Account balance is updated.

- Transaction logs are securely stored for record-keeping and auditing.
- Receipts are printed or sent digitally if requested.

Main Flow:

1. Customer inserts an ATM card or uses a cardless withdrawal option (e.g., QR code).
2. The ATM prompts the customer to enter their PIN.
3. Customer enters the PIN.
4. System verifies the PIN against the bank database. If valid, the customer proceeds.
5. ATM displays available transaction options; Customer selects **"Cash Withdrawal"**.
6. Customer enters the amount to withdraw.
7. System checks account balance and ATM cash availability.
   - If sufficient, it proceeds.
   - If insufficient, an error message is displayed.
8. If withdrawal is approved, ATM processes the transaction:
   - Deducts the amount from the customer's account.
   - Records the transaction for the bank and network processor.
   - Dispenses cash.
9. ATM offers the option to print a receipt or send a digital receipt via SMS/email.
10. ATM asks if the customer wants another transaction.
    - If **Yes**, the process starts over.
    - If **No**, the ATM ejects the card and displays a thank-you message.
11. Customer takes cash, receipt (if any), and leaves.

Alternate Flows:

a. Incorrect PIN Entered

1. Customer enters an incorrect PIN.
2. System notifies the customer and allows retries (up to 3 attempts).
3. After 3 failed attempts, ATM temporarily blocks the card and notifies the bank.

b. Insufficient Funds

1. Customer requests an amount exceeding the account balance.
2. System notifies the customer and prompts them to enter a lower amount.

c. ATM Out of Cash

1. If the ATM has insufficient cash, it notifies the customer.
2. Customers are given the option to withdraw a lower amount or cancel the transaction.

d. Card Retained Due to Security Reasons

1. If the ATM detects a blocked or expired card, it displays an error.
2. ATM retains the card and notifies the bank for security reasons.
3. Customers are advised to contact the bank.

e. Transaction Timeout

1. If the customer does not respond within a set time, the ATM cancels the transaction.
2. The card is ejected (if inserted).

Special Requirements:

- ATM must have **PCI DSS** security compliance for PIN and card data protection.
- The transaction must complete within **30 seconds** after PIN entry.
- ATM must support **multiple languages** based on location.
- Must allow **cardless withdrawals** via mobile authentication (e.g., OTP, QR code).
- **Emergency cash withdrawal** option for lost cards (if supported by the bank).
- **Camera monitoring** for fraud prevention and security.
- **Disaster recovery mechanism** to resume transactions in case of failures.

Technology and Data Variations:

- PIN entry via **touch screen, keypad, or biometric authentication**.
- Cardless withdrawal via **mobile banking app or NFC**.
- Receipt options:
    - **Printed via thermal printer**
    - **Digital receipt sent via SMS/email**
- Payment network interactions via **Visa, MasterCard, UPI, or other banking networks**.

3.Booking a Hotel Room

Use Case Number: HBK-US0001
Use Case Name: Booking a Hotel Room
Overview: A customer books a hotel room via an online portal or at the front desk. The system records the booking, processes payment, and confirms the reservation.

Primary Actor: Customer

Stakeholders and Interests:

- **Customer:** Wants a seamless booking process with accurate room details, pricing, and a confirmation receipt.
- **Hotel Receptionist:** Wants an easy way to check room availability and confirm bookings without delays.
- **Hotel Management:** Wants to ensure accurate records of all bookings and maximize room occupancy.
- **Payment Gateway Service:** Wants secure and accurate transaction processing.
- **Housekeeping Staff:** Wants real-time updates on room assignments and preparation schedules.
- **Government Tax Authorities:** Want proper tax collection on room bookings.

Preconditions: The customer is identified and authenticated (if booking online).

Postconditions: Booking is saved, tax is correctly calculated, payment is processed, and a confirmation receipt is generated.

Main Flow:

1. Customer visits the hotel website or approaches the reception desk.
2. Customer selects check-in and check-out dates.
3. System displays available rooms with pricing and amenities.
4. Customer selects a preferred room and enters personal details.

5. System calculates the total amount, including applicable taxes and service charges.
6. Customer confirms the booking and selects a payment method.
7. System processes payment and updates room availability.
8. System sends a booking confirmation via email/SMS and generates a receipt.
9. Customer receives confirmation details and completes the process.

Alternate Flows:

a. At any time, customer modifies or cancels the booking:

1. System retrieves booking details.
2. Customer selects modification or cancellation option.
3. System updates the booking, processes any applicable refunds, and sends confirmation.

b. System failure during booking process:

1. Customer is notified of the failure.
2. System attempts to recover the previous state.
3. If unsuccessful, customer starts a new booking.

c. Customer requests additional services (e.g., breakfast, airport pickup):

1. System displays add-on services.
2. Customer selects additional services.
3. System updates the booking and recalculates the total.

d. Payment fails:

1. System notifies customer.
2. Customer retries payment or selects a different payment method.
3. If multiple failures occur, the system may require alternative verification.

Special Requirements:

- User-friendly UI for desktop and mobile users.
- Booking confirmation should be sent within 30 seconds.

- Payment transactions must be logged for security and auditing.
- Multi-language support based on customer location.
- Data security compliance with GDPR and PCI DSS.

Technology and Data Variations:

- Room selection by category (single, double, suite).
- Payment method variations: Credit/Debit card, UPI, net banking, digital wallets.
- Confirmation via email, SMS, or in-app notification.