Student Name : Trần Quang Thái,Nguyễn Bảo Thắng

Student ID : 522K0037,522K0025

1. Clarify Ambiguous Requirements with Q&A Template

Questions to Clarify Requirements:

Number	Question
1	What types of public transportation tickets can be purchased (Bus, MRT, etc.)?
2	Can the user purchase multiple tickets in a single transaction?
3	Are there any discounts or promotions applied automatically?
4	What happens if the credit card transaction fails?
5	What types of digital wallets are supported for QR code payments?
6	Are receipts issued for transactions, and if so, how?
7	Is there a time limit for completing the payment once the QR code is shown?
8	What security measures are in place for handling credit card and digital wallet transactions?
9	How are destinations updated or maintained in the system?
10	Is there an option to cancel the transaction at any point?
11	Are there any user interface requirements for the menu display (e.g., languages, accessibility)?
12	What are the expected response times for each interaction (e.g., selecting destination, payment)?

2. Write Functional, Non-Functional, and Domain Requirements

Functional Requirements:

1. The system shall allow users to select a destination from a list of available options.
2. The system shall provide options for payment via credit card and digital wallet.
3. The system shall issue a paper ticket with a barcode when payment is made with a credit card.
4. The system shall display a QR code for digital wallet payments upon selecting a route.

5. The system shall validate credit card transactions before issuing the ticket.
6. The system shall update the user's account balance upon successful payment.
7. The system shall provide an option to cancel the transaction at any stage.
8. The system shall support multiple languages for user interaction.
9. The system shall print a receipt for each transaction.

Non-Functional Requirements:

1. The system shall have a response time of less than 2 seconds for displaying the menu of destinations.
2. The system shall complete credit card transactions within 5 seconds.
3. The system shall ensure secure handling of payment information.
4. The system shall be operational 24/7 with an uptime of 99.9%.
5. The system shall provide an intuitive and accessible user interface.
6. The system shall be capable of handling up to 100 transactions per hour without performance degradation.
7. The system shall log all transactions for audit purposes.

Domain Requirements:

1. The system shall comply with PCI-DSS standards for credit card transactions.
2. The system shall integrate with the public transportation ticketing database.
3. The system shall support commonly used digital wallets (e.g., Apple Pay, Google Pay).
4. The system shall be compliant with local regulations regarding electronic payments and data privacy.
5. The system shall interface with banking systems for payment verification.

3. Use Case Diagram and Descriptions

Use Case Diagram:

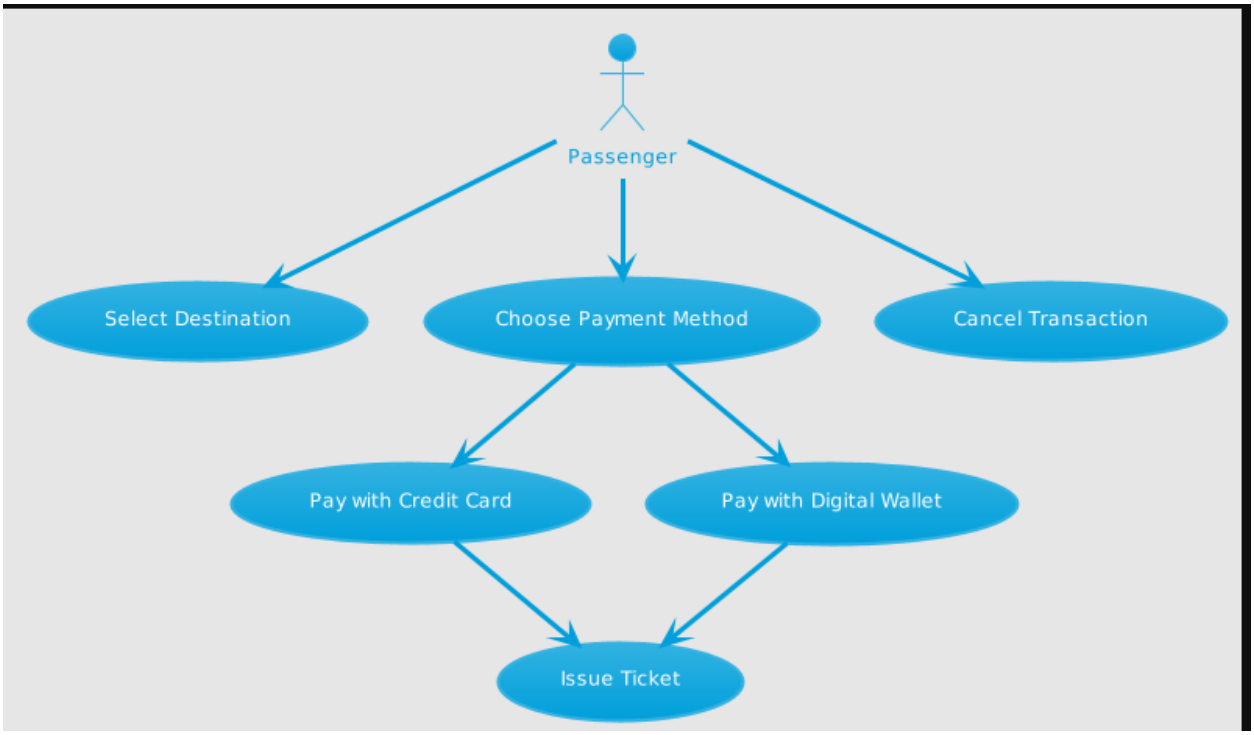
The use case diagram will include the following primary use cases:

- Select Destination
- Choose Payment Method
- Pay with Credit Card
- Pay with Digital Wallet
- Issue Ticket
- Cancel Transaction

Use Case Descriptions:

1. **Select Destination:**
    - **Actor:** Passenger
    - **Description:** The passenger selects the desired destination from a list of available destinations.
    - **Precondition:** The machine is operational, and the start button is pressed.
    - **Postcondition:** The destination is selected and ready for the next step.

2. **Choose Payment Method:**
    - **Actor:** Passenger
    - **Description:** The passenger selects the preferred method of payment (credit card or digital wallet).
    - **Precondition:** A destination has been selected.
    - **Postcondition:** The payment method is chosen, and the system is ready to process payment.
3. **Pay with Credit Card:**
    - **Actor:** Passenger
    - **Description:** The passenger inputs credit card details, and the system processes the payment.
    - **Precondition:** The payment method is selected as a credit card.
    - **Postcondition:** The credit card transaction is validated, and the ticket is issued.
4. **Pay with Digital Wallet:**
    - **Actor:** Passenger
    - **Description:** The system displays a QR code for the passenger to scan and complete the payment via a digital wallet.
    - **Precondition:** The payment method is selected as a digital wallet.
    - **Postcondition:** The payment is confirmed, and the ticket is issued.
5. **Issue Ticket:**
    - **Actor:** System
    - **Description:** The system issues a paper ticket with a barcode after a successful payment.
    - **Precondition:** Payment is successfully completed.
    - **Postcondition:** The ticket is printed and provided to the passenger.
6. **Cancel Transaction:**
    - **Actor:** Passenger
    - **Description:** The passenger can cancel the transaction at any stage before payment is confirmed.
    - **Precondition:** The transaction is in progress.
    - **Postcondition:** The transaction is canceled, and no charges are made.
7. **Use Case Diagram**



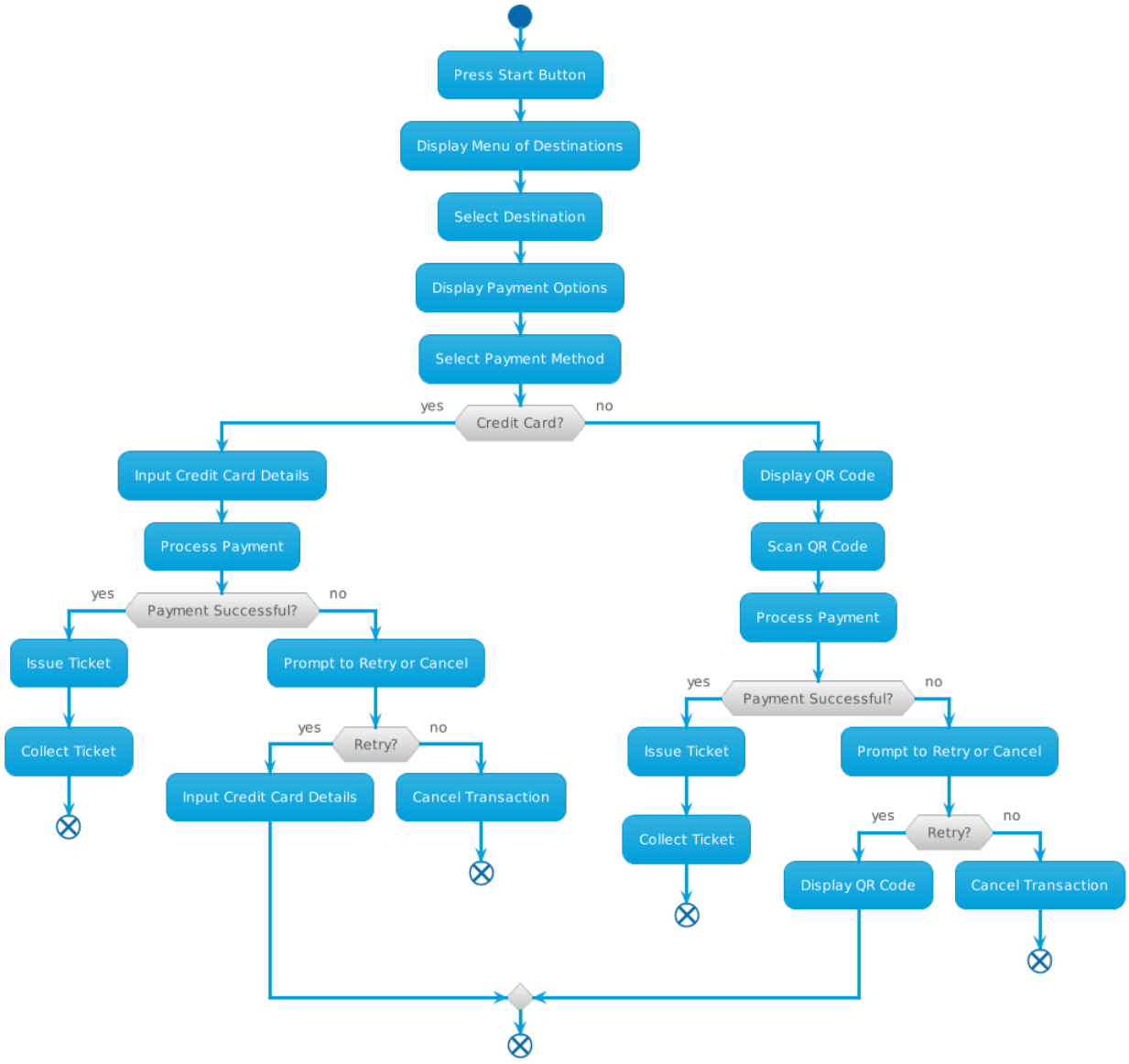
4. Activity Diagram

The activity diagram will outline the process of buying a ticket from the ticket vending machine.

Activity Diagram:

1. Passenger presses the start button.
2. System displays the menu of destinations.
3. Passenger selects a destination.
4. System displays payment options.
5. Passenger selects a payment method.
    - If credit card:
        1. Passenger inputs credit card.
        2. System processes payment.
        3. If payment successful, system issues ticket.
        4. If payment fails, system prompts to retry or cancel.
    - If digital wallet:
        1. System displays QR code.
        2. Passenger scans QR code.
        3. System processes payment.
        4. If payment successful, system issues ticket.
        5. If payment fails, system prompts to retry or cancel.
6. Passenger collects the ticket.

7. Transaction ends.

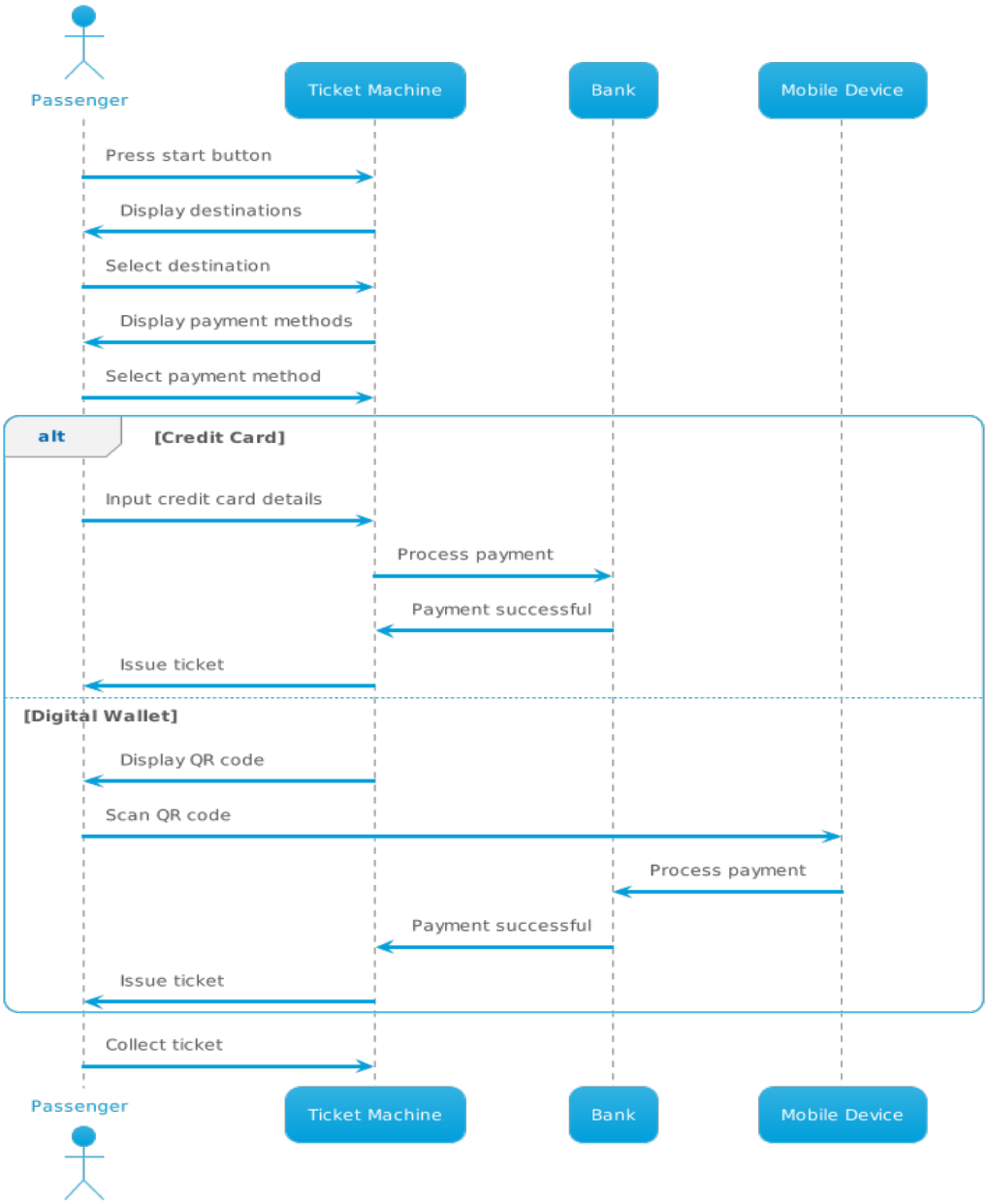


5. Sequence Diagram, State Chart Diagram, and Class Diagram

Sequence Diagram for "Buy a Ticket":

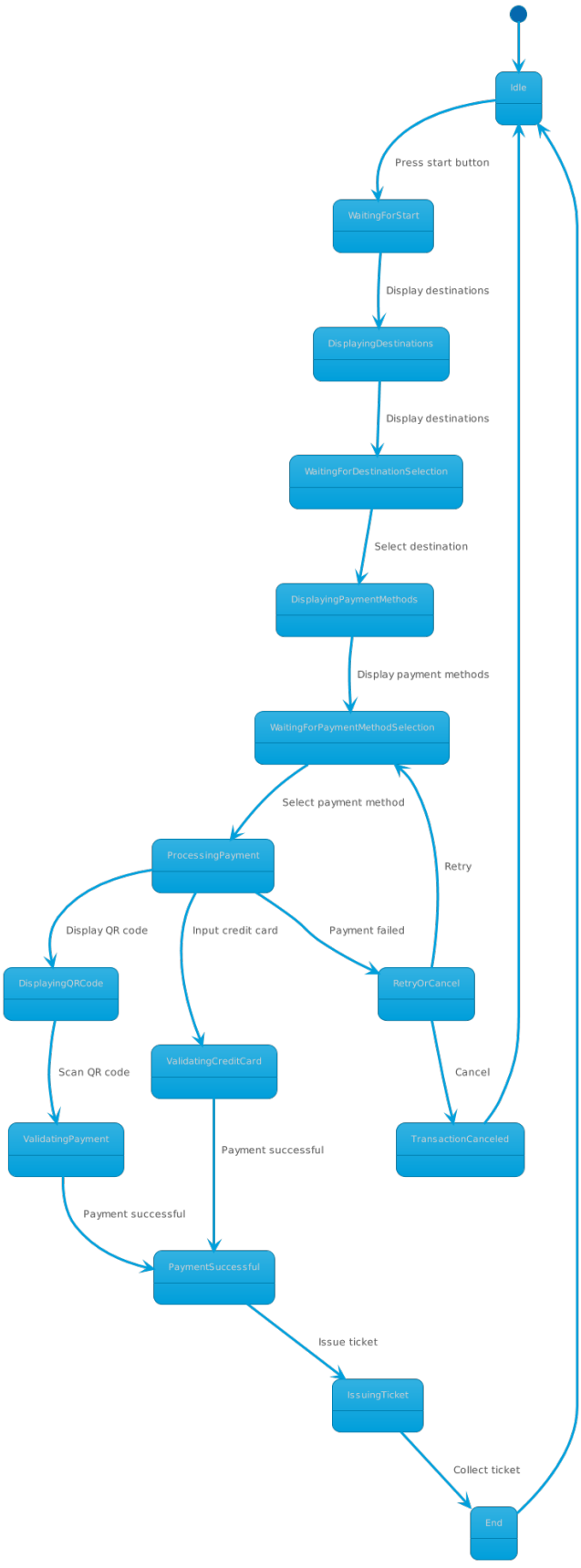
1. Passenger -> Ticket Machine: Press start button
2. Ticket Machine -> Passenger: Display destinations
3. Passenger -> Ticket Machine: Select destination
4. Ticket Machine -> Passenger: Display payment methods

5. Passenger -> Ticket Machine: Select payment method
    - If credit card:
        1. Passenger -> Ticket Machine: Input credit card
        2. Ticket Machine -> Bank: Process payment
        3. Bank -> Ticket Machine: Payment success/failure
        4. Ticket Machine -> Passenger: Issue ticket/Retry or cancel
    - If digital wallet:
        1. Ticket Machine -> Passenger: Display QR code
        2. Passenger -> Mobile Device: Scan QR code
        3. Mobile Device -> Bank: Process payment
        4. Bank -> Ticket Machine: Payment success/failure
        5. Ticket Machine -> Passenger: Issue ticket/Retry or cancel
6. Passenger -> Ticket Machine: Collect ticket



State Chart Diagram:

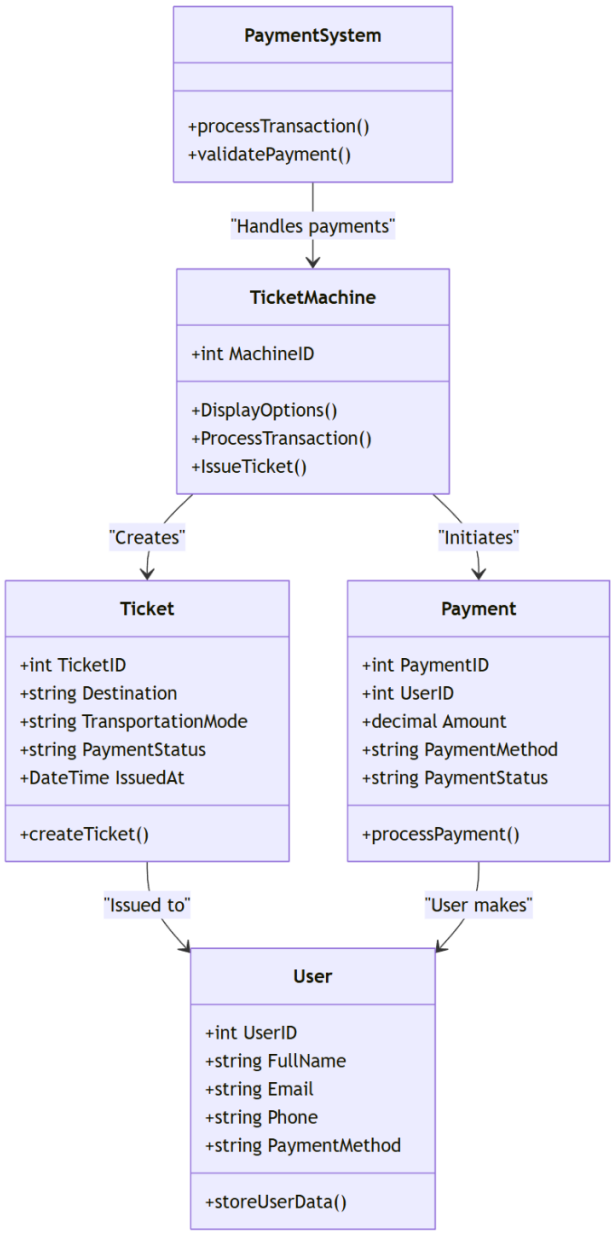
1. **Idle State:**
    - Transition to "Waiting for Start" on start button press.
2. **Waiting for Start:**
    - Transition to "Displaying Destinations" on start button press.
3. **Displaying Destinations:**
    - Transition to "Waiting for Destination Selection" on destination display.
4. **Waiting for Destination Selection:**
    - Transition to "Displaying Payment Methods" on destination selection.
5. **Displaying Payment Methods:**
    - Transition to "Waiting for Payment Method Selection" on payment method display.
6. **Waiting for Payment Method Selection:**
    - Transition to "Processing Payment" on payment method selection.
7. **Processing Payment:**
    - If credit card:
        - Transition to "Validating Credit Card" on input.
        - Transition to "Issuing Ticket" on successful validation.
        - Transition to "Retry or Cancel" on failure.
    - If digital wallet:
        - Transition to "Displaying QR Code" on selection.
        - Transition to "Validating Payment" on scan.
        - Transition to "Issuing Ticket" on successful validation.
        - Transition to "Retry or Cancel" on failure.
8. **Issuing Ticket:**
    - Transition to "Transaction Complete" on ticket issue.
9. **Retry or Cancel:**
    - Transition to "Waiting for Payment Method Selection" on retry.
    - Transition to "Transaction Canceled" on cancel.
10. **Transaction Complete:**
    - Transition to "Idle State" on completion.
11. **Transaction Canceled:**
    - Transition to "Idle State" on cancellation.


Idle
Press start button
WaitingForStart
Display destinations
DisplayingDestinations
Display destinations
WaitingForDestinationSelection
Select destination
DisplayingPaymentMethods
Display payment methods
WaitingForPaymentMethodSelection
Select payment method
ProcessingPayment
Retry
Display QR code
Input credit card
Payment failed
DisplayingQRCode
RetryOrCancel
ValidatingCreditCard
Scan QR code
Cancel
ValidatingPayment
Payment successful
TransactionCanceled
Payment successful
PaymentSuccessful
Issue ticket
IssuingTicket
Collect ticket
End

Class Diagram:

Classes:

1. **TicketMachine**
    - Attributes: id, location, status
    - Methods: start(), displayDestinations(), displayPaymentMethods(), issueTicket(), cancelTransaction()
2. **Passenger**
    - Attributes: id, currentTransaction
    - Methods: selectDestination(), selectPaymentMethod(), cancelTransaction()
3. **Payment**
    - Attributes: type, amount, status
    - Methods: processPayment(), validatePayment()
4. **CreditCardPayment (inherits Payment)**
    - Attributes: cardNumber, expiryDate, cvv
    - Methods: validateCard(), chargeCard()
5. **DigitalWalletPayment (inherits Payment)**
    - Attributes: walletID
    - Methods: generateQRCode(), validateQRCode()
6. **Ticket**
    - Attributes: ticketNumber, destination, barcode, status
    - Methods: printTicket(), validateTicket()



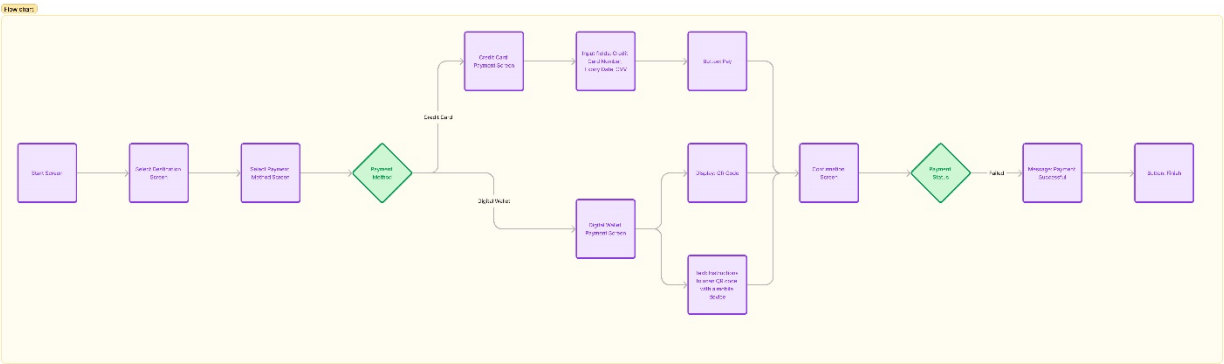
6. Wireframe/Mockup Design for Use Cases

Key Screens:

1. **Start Screen**
    - Button: "Start"
2. **Select Destination Screen**

- List of destinations (e.g., buttons or dropdown)
- Button: "Next"

3. **Select Payment Method Screen**
   - Radio buttons or dropdown: "Credit Card" or "Digital Wallet"
   - Button: "Next"
4. **Credit Card Payment Screen**
   - Input fields: "Credit Card Number", "Expiry Date", "CVV"
   - Button: "Pay"
5. **Digital Wallet Payment Screen**
   - Display: QR Code
   - Text: Instructions to scan QR code with a mobile device
6. **Confirmation Screen**
   - Message: "Payment Successful" or "Payment Failed"
   - Button: "Finish"



7. Architecture Design and Deployment Diagram

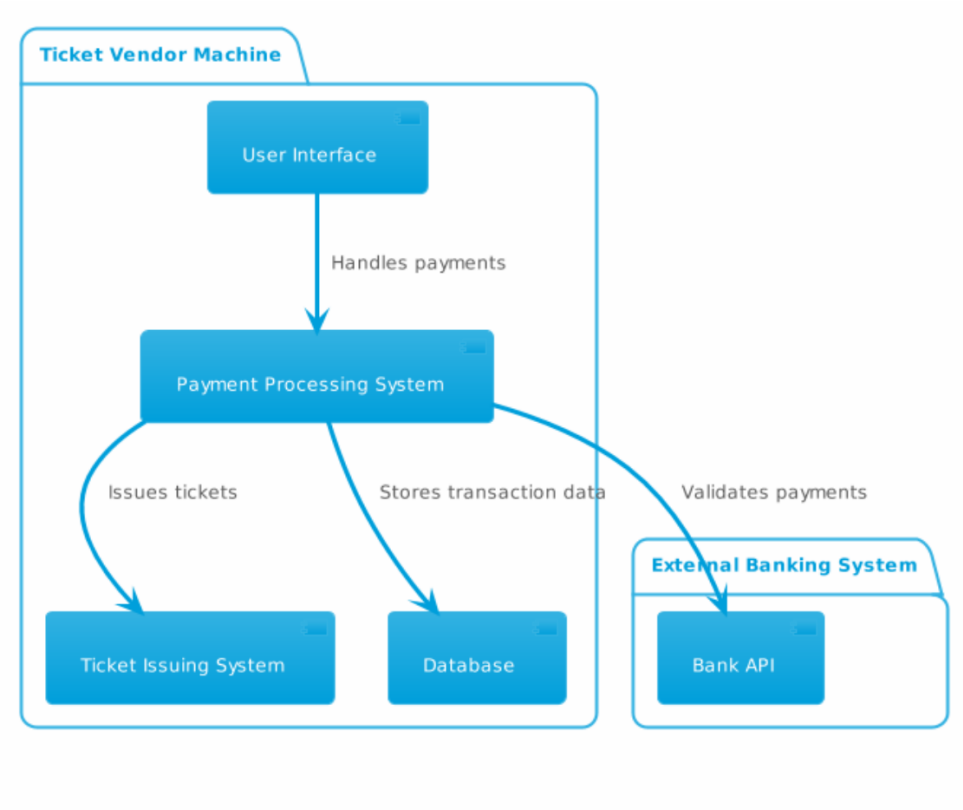
Architecture Design

Components:

1. **User Interface (UI)**
   - Front-end application running on the ticket machine
   - User interactions for selecting destination and payment method
2. **Payment Processing System**
   - Handles credit card and digital wallet payments
   - Connects to external banking systems
3. **Ticket Issuing System**
   - Generates and prints the ticket with a barcode
4. **Database**
   - Stores transaction logs, payment details, and ticket information

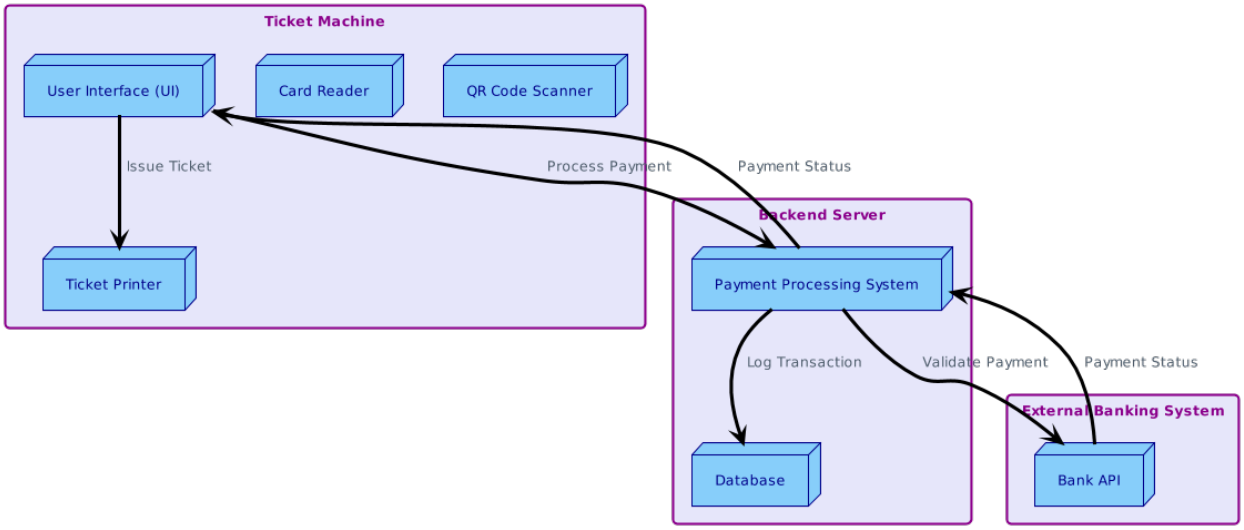
Interactions:

- UI interacts with the Payment Processing System for handling payments
- Payment Processing System communicates with external banking systems
- Ticket Issuing System gets triggered upon successful payment
- Database logs each transaction and payment status



Deployment Diagram

1. **Ticket Machine (Client-Side)**
   - Runs the front-end application
   - Includes components: UI, Card Reader, QR Code Scanner, Ticket Printer
2. **Server-Side (Backend)**
   - Web server hosting the payment processing application
   - Database server for storing transaction data
3. **External Banking System**
   - For processing credit card and digital wallet transactions


Ticket Machine
User Interface (UI)
Card Reader
QR Code Scanner
Issue Ticket
Ticket Printer
Process Payment
Payment Status
Backend Server
Payment Processing System
Log Transaction
Validate Payment
Payment Status
Database
External Banking System
Bank API