MAHATMA EDUCATION SOCIETY’S

PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE
(Autonomous)

NEW PANVEL

PROJECT REPORT ON

“Bus Ticket Booking System”

IN PARTIAL FULFILLMENT OF

BACHELOR OF Science Information Technology

SEMESTER IV– 2023-24

PROJECT GUIDE

Anju Ma’am

SUBMITTED BY: Mehtab Singh Randhawa

ROLL NO: 6195

Introduction:

The Bus Ticket Booking System in Java Swing is an application designed to facilitate the booking of bus tickets. It provides a user-friendly interface for customers to enter their details, select their destination and pickup point, choose the number of tickets, and make payment using various methods such as UPI, bank transfer, or debit/credit card. The system ensures data validation by verifying inputs such as the name, date format, and payment details. Additionally, it calculates the total amount to be paid based on the selected destination and number of tickets.

Explanation:

1. **Import Statements:** The code begins with importing necessary Java Swing libraries for GUI components and event handling.

2. **BusBookingSystem Class:** This class extends JFrame and implements ActionListener, allowing it to create a window for the bus ticket booking system and handle events such as button clicks.

3. **Instance Variables:**
    - nameField, dateField: JTextField instances for entering passenger name and date of booking.
    - destinationComboBox, pickupComboBox, paymentMethodComboBox: JComboBox instances for selecting destination, pickup point, and payment method.
    - ticketSpinner: JSpinner instance for selecting the number of tickets.
    - bookButton: JButton instance for booking the ticket.

4. **Constructor:**
    - Sets the title, size, default close operation, and location of the frame.
    - Initializes a JPanel and sets its layout to null.
    - Creates and adds labels, combo boxes, text fields, spinner, and button to the panel.
5. **ActionListener Implementation:**
    - Implements the actionPerformed method to handle button click events.
    - Retrieves input values from the GUI components.
    - Validates input data (name, date format, bank account/card number).
    - Calculates the total amount to be paid based on the selected destination and number of tickets.
    - Displays a message dialog with the total amount to be paid.
    - Depending on the selected payment method, prompts the user to enter additional details (UPI ID, bank account number, card details) and displays a success message if the payment is successful.

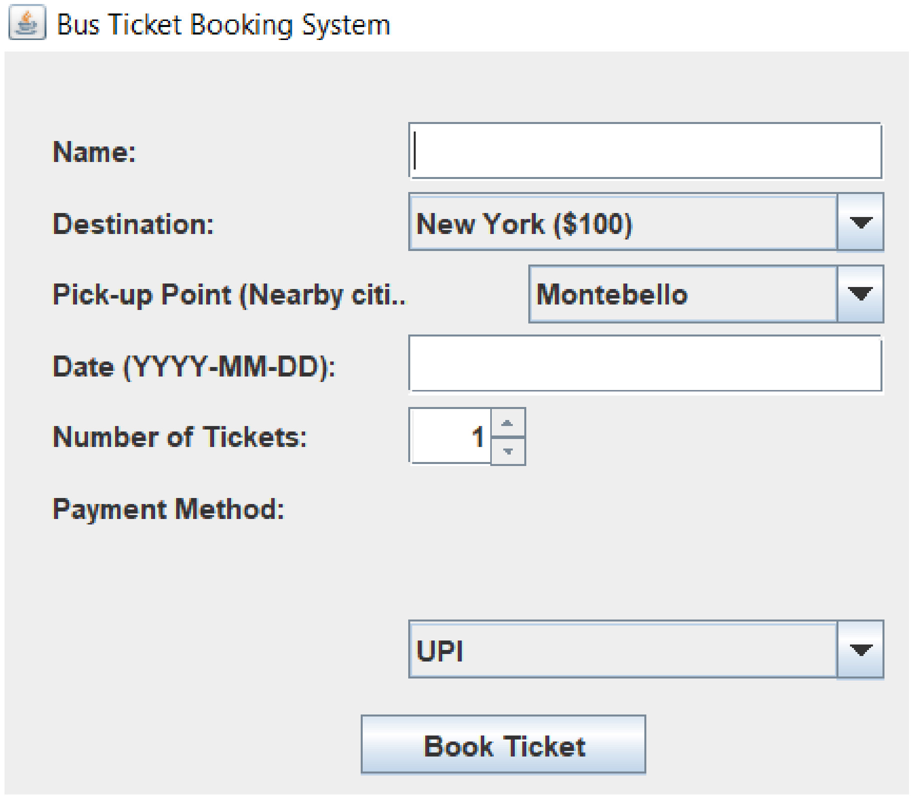
6. **Helper Methods:**
    - isValidNumber: Checks if a string consists of only digits.
    - isValidExpiryDate: Checks if a string represents a valid expiry date in the format "MM/YY".
    - getPrice: Returns the price based on the selected destination.

7. **Main Method:**
    - Invokes the BusBookingSystem constructor using the SwingUtilities.invokeLater method to ensure the GUI components are created and modified on the Event Dispatch Thread (EDT).

Code:

```
import javax.swing.*;
import java.awt.event.*;
public class BusBookingSystem extends JFrame implements ActionListener
{
    private JTextField nameField, dateField;
    private JComboBox<String> destinationComboBox, pickupComboBox,
paymentMethodComboBox;
    private JSpinner ticketSpinner;
    private JButton bookButton;

    public BusBookingSystem()
    {
        setTitle("Bus Ticket Booking System");
        setSize(400, 350);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel();
        add(panel);
        panel.setLayout(null);

        JLabel[] labels =
        {
            new JLabel("Name:"), new JLabel("Destination:"), new JLabel("Pick-up Point
(Nearby cities):"),
            new JLabel("Date (YYYY-MM-DD):"), new JLabel("Number of Tickets:"), new
JLabel("Payment Method:")
        };
        for (int i = 0; i < labels.length; i++)
        {
            JLabel label = labels[i];
            label.setBounds(20, 30 + 30 * i, 150, 25);
            panel.add(label);
        }
        String[] destinations = {"New York ($100)", "Los Angeles ($80)", "Chicago ($90)",
"Houston ($70)"};
        destinationComboBox = new JComboBox<>(destinations);
        destinationComboBox.setBounds(170, 60, 200, 25);
        panel.add(destinationComboBox);
        String[] pickups = {"Montebello", "Green Island", "Mexico"};
        pickupComboBox = new JComboBox<>(pickups);
        pickupComboBox.setBounds(220, 90, 150, 25);
        panel.add(pickupComboBox);

        paymentMethodComboBox = new JComboBox<>(new String[]{"UPI", "Pay by
Bank", "Debit or Credit Card"});
        paymentMethodComboBox.setBounds(170, 240, 200, 25);
        panel.add(paymentMethodComboBox);

        nameField = new JTextField(20);
        nameField.setBounds(170, 30, 200, 25);
        panel.add(nameField);

        dateField = new JTextField(20);
        dateField.setBounds(170, 120, 200, 25);
        panel.add(dateField);

        ticketSpinner = new JSpinner(new SpinnerNumberModel(1, 1, 10, 1));
        ticketSpinner.setBounds(170, 150, 50, 25);
        panel.add(ticketSpinner);

        bookButton = new JButton("Book Ticket");
        bookButton.setBounds(150, 280, 120, 25);
        bookButton.addActionListener(this);
        panel.add(bookButton);

        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource() == bookButton)
           {
            String name = nameField.getText(), date = dateField.getText();
            String destination = (String) destinationComboBox.getSelectedItem();
            String pickup = (String) pickupComboBox.getSelectedItem();
            if (!name.matches("[a-zA-Z]+") || !date.matches("\\d{4}-\\d{2}-\\d{2}"))
           {
                JOptionPane.showMessageDialog(this, "Invalid input!", "Error",
JOptionPane.ERROR_MESSAGE);
                return;
            }
            double price = getPrice(destination);
            int tickets = (int) ticketSpinner.getValue();
            double totalAmount = price * tickets;

            JOptionPane.showMessageDialog(this, "Total amount to be paid: $" +
totalAmount, "Total Amount", JOptionPane.INFORMATION_MESSAGE);

            String paymentMethod = (String) paymentMethodComboBox.getSelectedItem();
            switch (paymentMethod)
            {
                case "UPI":
                    String upiId = JOptionPane.showInputDialog(this, "Enter your UPI ID:");
                    if (upiId != null)
                    {
                        JOptionPane.showMessageDialog(this, "Payment successful with UPI.",
"Success", JOptionPane.INFORMATION_MESSAGE);
                    }
                    break;

                case "Pay by Bank":
                    String bankAccount = JOptionPane.showInputDialog(this, "Enter your bank
account number:");
                    if (bankAccount != null && isValidNumber(bankAccount))
                    {
                        JOptionPane.showMessageDialog(this, "Payment successful via Bank.",
"Success", JOptionPane.INFORMATION_MESSAGE);
                    }
                    else
                    {
                        JOptionPane.showMessageDialog(this, "Invalid bank account number.",
"Error", JOptionPane.ERROR_MESSAGE);
                    }
                    break;
                case "Debit or Credit Card":
                    String cardNumber = JOptionPane.showInputDialog(this, "Enter your card
number:");
                    String expiryDate = JOptionPane.showInputDialog(this, "Enter your card
expiry date (MM/YY):");
                    if (cardNumber != null && expiryDate != null &&
isValidNumber(cardNumber) && isValidExpiryDate(expiryDate))
                    {
                        JOptionPane.showMessageDialog(this, "Payment successful with card.",
"Success", JOptionPane.INFORMATION_MESSAGE);
                    }
                    else
                    {
                        JOptionPane.showMessageDialog(this, "Invalid card details.", "Error",
JOptionPane.ERROR_MESSAGE);
                    }
                    break;
            }
        }
    }
    private boolean isValidNumber(String number)
    {
        return number.matches("\\d+");
    }
    private boolean isValidExpiryDate(String expiryDate)
    {
        return expiryDate.matches("\\d{2}/\\d{2}");
    }
    private double getPrice(String destination)
    {
        if (destination.contains("New York")) return 100.00;
        if (destination.contains("Los Angeles")) return 80.00;
        if (destination.contains("Chicago")) return 90.00;
        if (destination.contains("Houston")) return 70.00;
        return 0.00;
    }
    public static void main(String[] args)
    {   SwingUtilities.invokeLater(BusBookingSystem::new);
    }
}
```

Output:



Bus Ticket Booking System

Name:

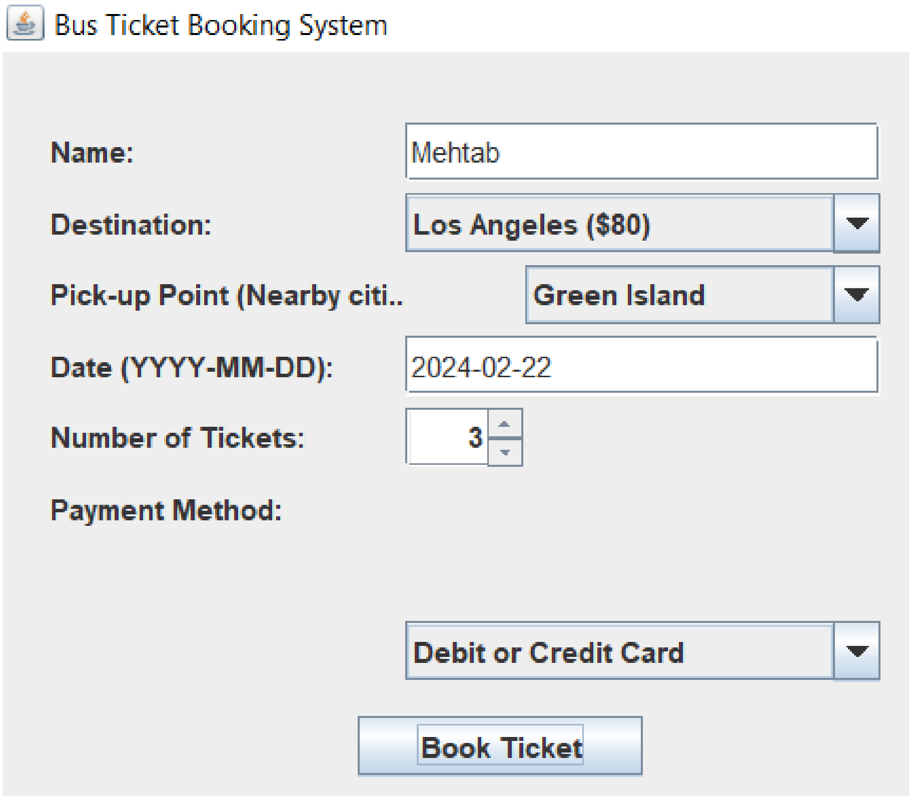
Destination: New York ($100)

Pick-up Point (Nearby citi.. Montebello

Date (YYYY-MM-DD):

Number of Tickets: 1

Payment Method: UPI



Bus Ticket Booking System

Name: Mehtab

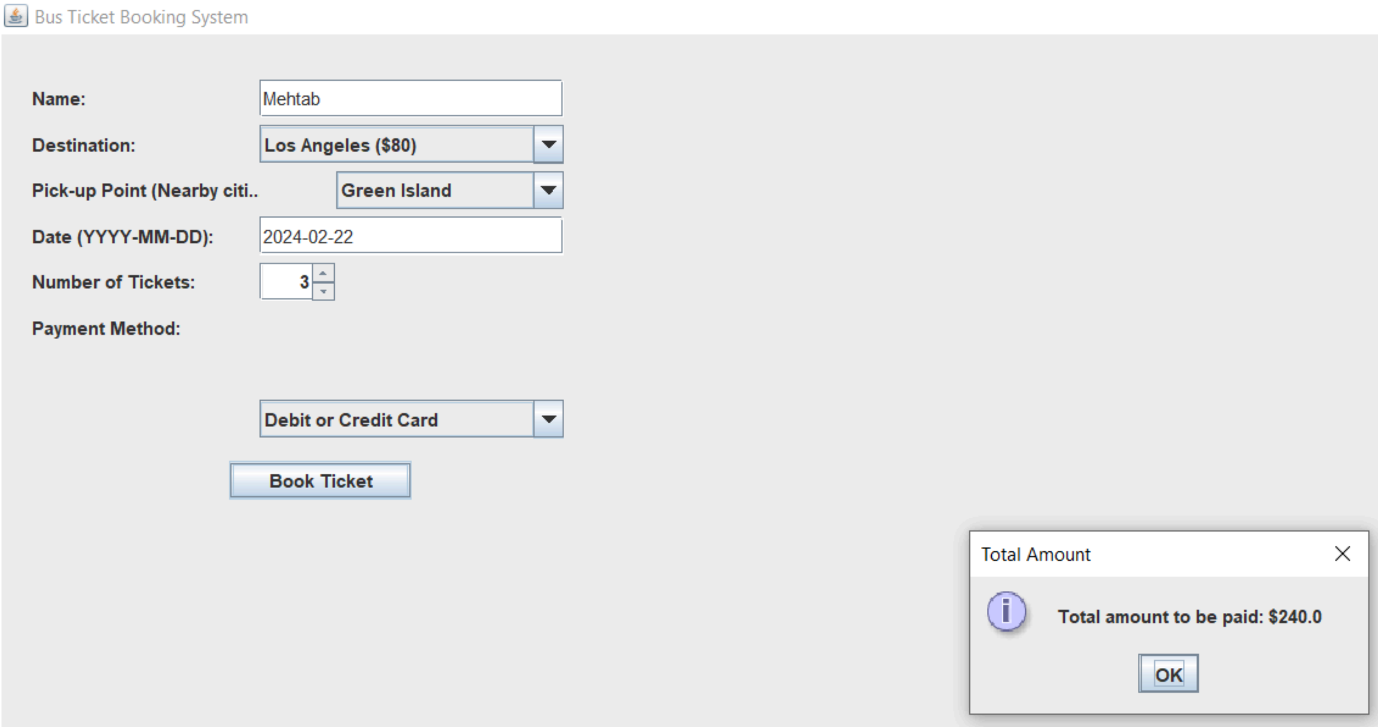
Destination: Los Angeles ($80)

Pick-up Point (Nearby citi.. Green Island

Date (YYYY-MM-DD): 2024-02-22

Number of Tickets: 3

Payment Method: Debit or Credit Card


Bus Ticket Booking System

Name: Mehtab

Destination: Los Angeles ($80)

Pick-up Point (Nearby citi.. Green Island

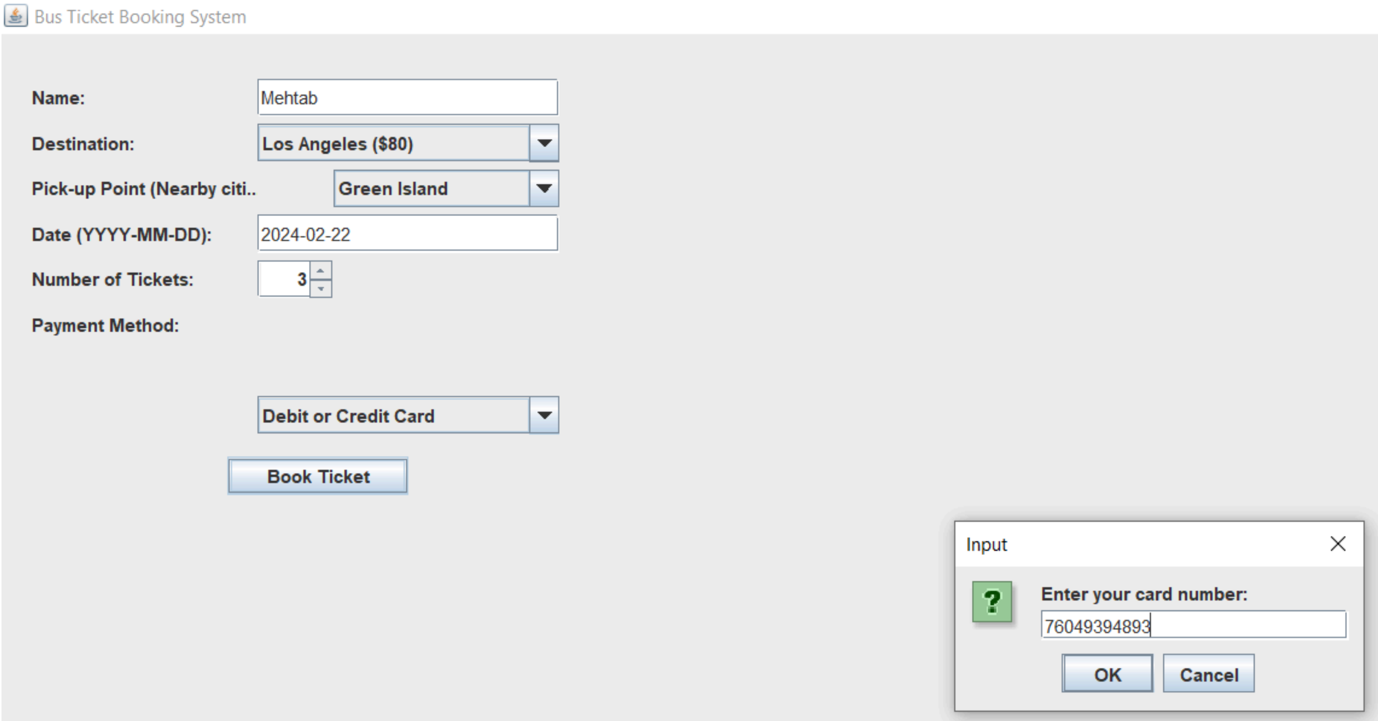
Date (YYYY-MM-DD): 2024-02-22

Number of Tickets: 3

Payment Method: Debit or Credit Card

Total Amount

Total amount to be paid: $240.0


Bus Ticket Booking System

Name: Mehtab

Destination: Los Angeles ($80)

Pick-up Point (Nearby citi.. Green Island

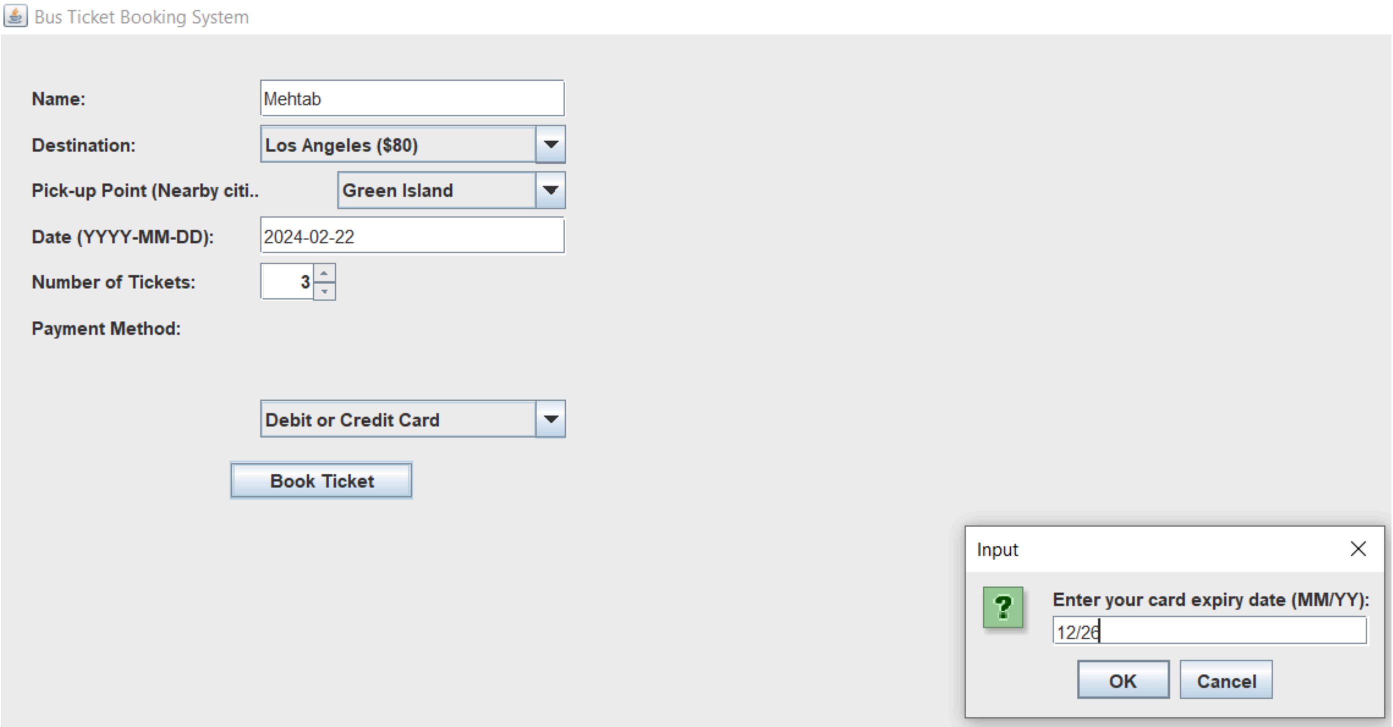
Date (YYYY-MM-DD): 2024-02-22

Number of Tickets: 3

Payment Method: Debit or Credit Card

Input

Enter your card number: 76049394893



Name: Mehtab

Destination: Los Angeles ($80)

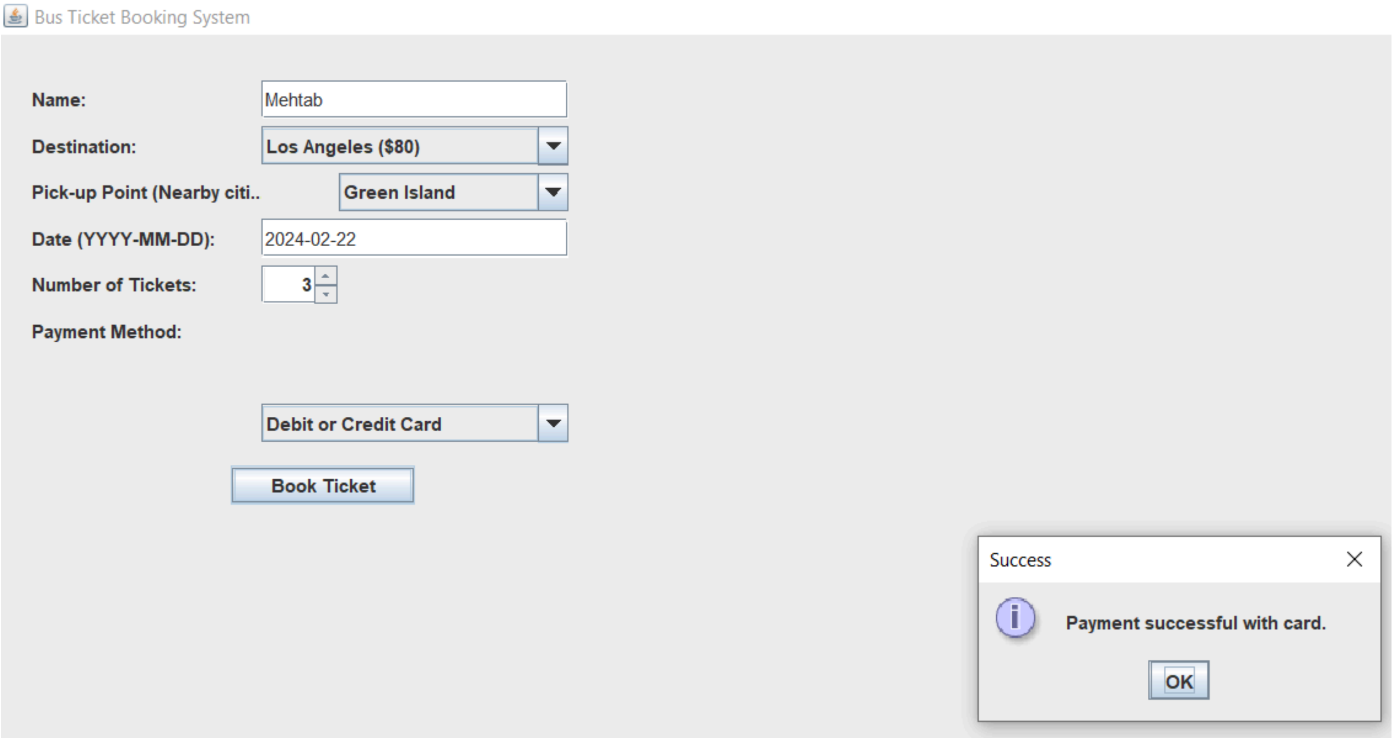
Pick-up Point (Nearby citi.. Green Island

Date (YYYY-MM-DD): 2024-02-22

Number of Tickets: 3

Payment Method: Debit or Credit Card

Enter your card expiry date (MM/YY): 12/26



Name: Mehtab

Destination: Los Angeles ($80)

Pick-up Point (Nearby citi.. Green Island

Date (YYYY-MM-DD): 2024-02-22

Number of Tickets: 3

Payment Method: Debit or Credit Card

Payment successful with card.

Conclusion:

In conclusion, the Bus Ticket Booking System offers a convenient and efficient way for customers to book bus tickets with ease. By providing a simple and intuitive interface, users can quickly enter their information, select their travel preferences, and complete the booking process securely. The system's features, including data validation and payment options, enhance the overall user experience and streamline the ticket booking process. With its robust functionality and user-friendly design, the Bus Ticket Booking System serves as a valuable tool for both customers and bus service providers.