```
# Author: [Rajeem Davis]
# Date Created: [December 1, 2023.]
# Course: ITT103
# Purpose: This program provides  is an  automated reservation system for UCC Signature Express Limited, enabling customers to book seats

class ReservationSystem:
    def __init__(self):
        self.first_class_rows = 5
        self.first_class_columns = 6
        self.business_class_rows = 6
        self.business_class_columns = 7
        self.economy_class_rows = 8
        self.economy_class_columns = 7

        self.first_class_seats = [['0'] * self.first_class_columns for _ in range(self.first_class_rows)]
        self.business_class_seats = [['0'] * self.business_class_columns for _ in range(self.business_class_rows)]
        self.economy_class_seats = [['0'] * self.economy_class_columns for _ in range(self.economy_class_rows)]

    def display_menu(self):
        print("UCC Signature Express Limited")
        print("Come one come all and book your reservations and enjoy the ride!")
        print("Reservation Options:")
        print("▯ First Class (F/f)")
        print("▯ Business Class (B/f)")
        print("▯ Economy Class (E/e)")
        print("▯ Quit or Cancel (Q/q)")

    def reserve_seat(self, class_type):
        while True:
            self.display_menu()
            choice = input("Please select an option: ")

            if choice == 'q':
                print(f"Reservation Type: {class_type} Class")
                print(f"Total number of seats: {self.get_total_seats(class_type)}")
                print(f"Total number of seats reserved: {self.get_reserved_seats(class_type)}")
                break

            elif choice in {'f', 'b', 'e'}:
                row = int(input("Enter row number (positive integer): "))
                column = int(input("Enter column number (positive integer): "))

                if row <= 0:
                    print("Number must be positive or greater than zero!")
                    continue

                if self.is_seat_available(class_type, row, column):
                    self.reserve_in_class(class_type, row, column)
                    print(f"Reserving seat: row {row} column {column:02}")
                else:
                    print("Seat already reserved! Please choose another seat.")

            else:
                print("Invalid choice!")

    def is_seat_available(self, class_type, row, column):
        seats = self.get_seats(class_type)
        return seats[row - 1][column - 1] == '0'

    def reserve_in_class(self, class_type, row, column):
        seats = self.get_seats(class_type)
        seats[row - 1][column - 1] = '1'

    def get_seats(self, class_type):
        if class_type == 'f':
            return self.first_class_seats
        elif class_type == 'b':
            return self.business_class_seats
        elif class_type == 'e':
            return self.economy_class_seats

    def get_total_seats(self, class_type):
        rows, columns = self.get_dimensions(class_type)
        return rows * columns

    def get_reserved_seats(self, class_type):
        seats = self.get_seats(class_type)
        return sum(row.count('1') for row in seats)

    def get_dimensions(self, class_type):
        if class_type == 'f':
```

```
            return self.first_class_rows, self.first_class_columns
        elif class_type == 'b':
            return self.business_class_rows, self.business_class_columns
        elif class_type == 'e':
            return self.economy_class_rows, self.economy_class_columns

if __name__ == "__main__":
    reservation_system = ReservationSystem()

    while True:
        class_choice = input("Enter the class (First Class F/f, Business Class B/b, Economy Class E/e) or Quit (Q/q): ")

        if class_choice in {'f', 'b', 'e'}:
            reservation_system.reserve_seat(class_choice)
        elif class_choice == 'q':
            print("You are exiting the reservation system. Thanks and Goodbye!")
            break
        else:
            print("Invalid choice. Please enter F/f, B/b, E/e, or Q/q.")
```