



# DATA STRUCTURE & ALGORITHM PROJECT

*Presented by: GUSION*

*Che Marhumi*

*Iman Firdaus*

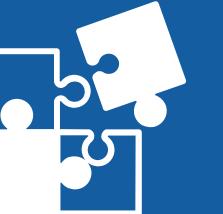
*Ariff Danish*

# AIRLINE RESERVATION SYSTEM

# PROJECT DESCRIPTION

The objective of this project is to develop our own model of the Airline Reservation System. This system's aim is to ease the process for booking/reserving purposes for customers, while for administrator or staff is to validate the customers request, update or remove the customer from the booking list, search for customers and get information of customers when displaying the list with the user-friendly menu. The main idea and implementation of this project is to develop the system using the data structure concept of queue and must be correctly applied using C++ language. In this project we implemented the queue using a linked list.

- Customers successfully request for the booking(enqueue)
- Customers can undo the request for reservation(dequeue)
- Admin can validate or approve the request reservation or booking from customer(dequeue from request list then enqueue to booking list)
- Admin can view details of customers list
- Admin can remove customer from booking list(dequeue)
- Admin can search for customer using IC number
- System provides user-friendly menu driven



# PROJECT DESIGN

# PSEUDOCODE

Here's a pseudocode for the main part of our program in project:

```

```plaintext
class Customer
    attributes: name, destination, airline, day, month, year, IC

class BookingQueue
    class Node
        attributes: data (Customer), next (Node)

    attributes: front (Node), rear (Node)

    method enqueue(data)
        create Node with data
        if isEmpty()
            front = rear = newNode
        else
            rear.next = newNode
            rear = newNode

    method dequeue()
        if not isEmpty()
            temp = front
            front = front.next
            delete temp

    method isEmpty()
        return front == null

    method getFront()
        if not isEmpty()
            return front.data
        else
            throw out_of_range("Queue is empty")

class BookingSystem
    attributes: doneBooking (BookingQueue), reservationRequests (BookingQueue)

    method addBooking(booking)
        doneBooking.enqueue(booking)

    method reserveBooking(booking)
        reservationRequests.enqueue(booking)
```

```

```

method cancelRecentReservation()
if not reservationRequests.isEmpty()
    reservationRequests.dequeue()
    print "Most recent reservation canceled."
    return true
else
    print "No reservation requests to cancel."
    return false

method approveReservation()
if not reservationRequests.isEmpty()
    process all reservations
    print "All reservations approved."
else
    print "No reservation requests to approve."

method searchByICNumber(icNumber)
for each booking in doneBooking
    if booking.getICNum() == icNumber
        display booking details
        return true
print "No bookings found for IC Number " + icNumber
return false

method removeBookingByIC(icNumber)
find and remove booking by IC number
print "Booking for IC Number " + icNumber + " removed."

method displayAllBookings()
display all bookings

method displayRequestedBookings()
display reservation requests

function extractData(bookingSystem)
open "booking.txt"
if file not found
    print "Cannot open file."
    return
else
    read lines from file
    extract and add data to bookingSystem

function customerMenu(bookingSystem)
do
    display menu
    read user choice
```

```

```

switch user choice
case 1:
    input customer details
    create Customer
    bookingSystem.reserveBooking(newCustomer)
    print "Reservation request sent."
case 2:
    input IC number
    bookingSystem.cancelRecentReservation()
case 3:
    print "Returning to main menu."
while user choice is not 3

function adminMenu(bookingSystem)
do
    display menu
    read user choice
    switch user choice
        case 1:
            bookingSystem.approveReservation()
        case 2:
            bookingSystem.displayAllBookings()
        case 3:
            print "Returning to main menu."
        case 4:
            input IC number
            bookingSystem.searchByICNumber(IC number)
        case 5:
            input IC number
            bookingSystem.removeBookingByIC(IC number)
    while user choice is not 3

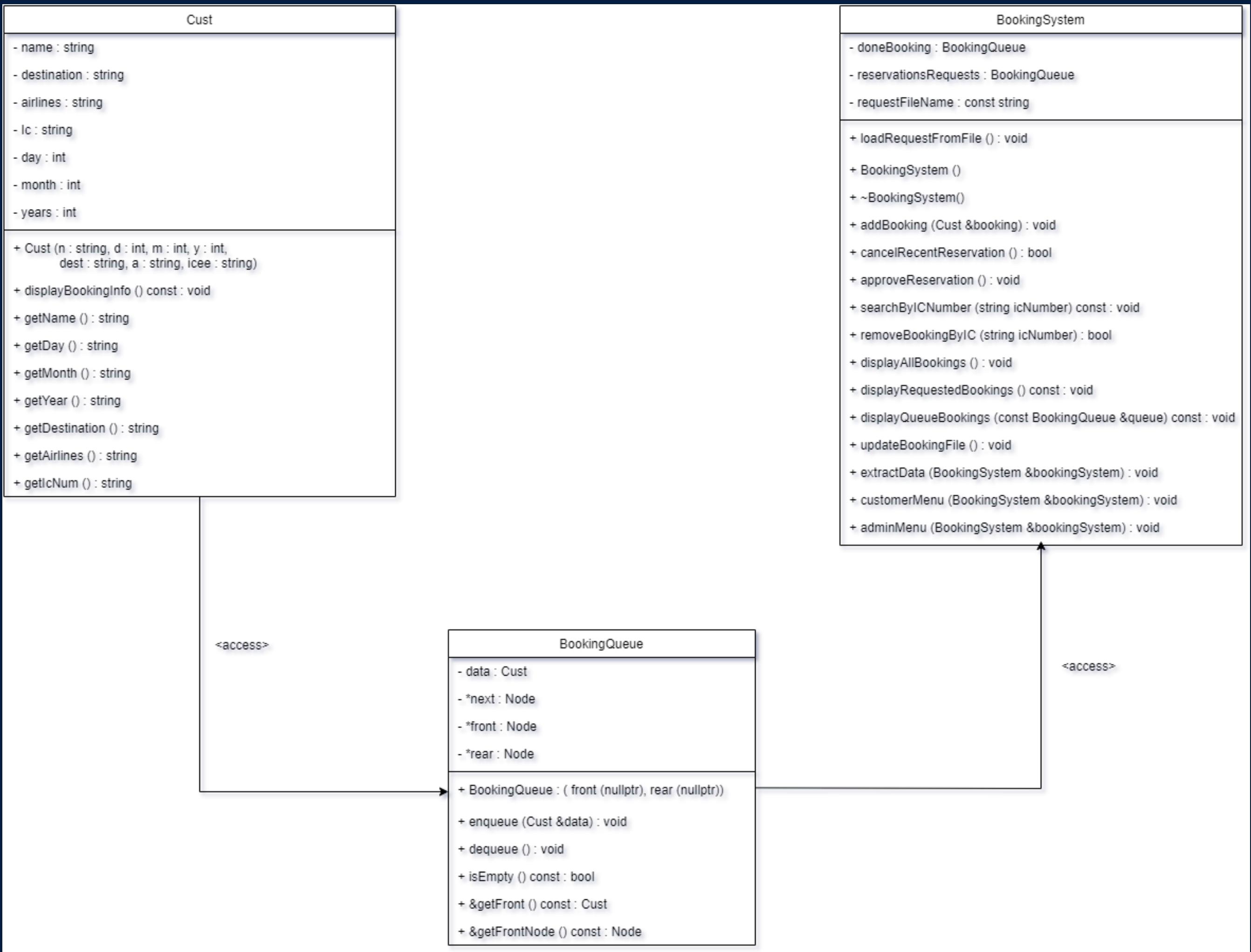
function main()
create BookingSystem
call extractData(bookingSystem)

do
    display main menu
    read user type choice
    switch user type choice
        case 1:
            call adminMenu(bookingSystem)
        case 2:
            call customerMenu(bookingSystem)
    while user type choice is not 3

pause and return 0
```

```

# CLASS DIAGRAM



# **DATA STRUCTURE CONCEPT EMPLOYED & EXECUTION OF THE PROGRAM**

# ENQUEUE OPERATION

```
Enter your choice: 2

+===== CUSTOMER MENU =====+
+1. Request Reservation      +
+2. Cancel Reservation (Undo) +
+3. Back to Main Menu        +
+=====

Enter your choice: 1
Enter customer name: Harun Haziq Bin Rizki

Enter date (day month year):
Day : 20
Month :12
Year :2025

Enter destination: KUL-JHB
Enter airline: MasAirline
Enter IC number: 030423030779

Reservation request sent successfully.
```

```
void enqueue(Cust &data)
{
    Node *newNode = new Node{data, nullptr};
    if (isEmpty())
    {
        front = rear = newNode;
    }
    else
    {
        rear->next = newNode;
        rear = newNode;
    }
}
```

# DEQUEUE OPERATION

```
Enter your choice: 5
Enter customer IC number to remove from booking: 30412

Booking for IC Number 30412 removed successfully.
```

```
+===== ADMIN MENU =====+
+1. Approve Reservation +
+2. Display All Bookings +
+3. Back to Main Menu +
+4. Search Customer +
+5. Remove Booking +
+=====
```

| Name       | Date of Booking | Destination | Airlines   | IC Number    |
|------------|-----------------|-------------|------------|--------------|
| Iman       | 26 / 12 / 2003  | KB-KK       | MasAirline | 20103        |
| Marhumi    | 16 / 6 / 2009   | PNG-KL      | Malindo    | 93412        |
| Ahmad Labu | 2 / 23 / 2005   | KUL-POL     | MasAirline | 030423030779 |
| Danish     | 31 / 12 / 2030  | KUL-JHB     | MasAirline | 030423030779 |

```
void dequeue()
{
    if (!isEmpty())
    {
        Node *temp = front;
        front = front->next;
        delete temp;
    }
}
```

# SEARCHING OPERATION

Enter your choice: 4

Enter customer IC number to search: 30412

## Booking details for IC Number 30412:

| Name  | Date of Booking | Destination | Airlines | IC Number |
|-------|-----------------|-------------|----------|-----------|
| Ariff | 15 / 5 / 2008   | KT-AS       | AirAsia  | 30412     |

# OTHERS: DISPLAY

| Enter your choice: 2 |                 |             |            |              |
|----------------------|-----------------|-------------|------------|--------------|
| Name                 | Date of Booking | Destination | Airlines   | IC Number    |
| Iman                 | 26 / 12 / 2003  | KB-KK       | MasAirline | 20103        |
| Ariff                | 15 / 5 / 2008   | KT-AS       | AirAsia    | 30412        |
| Marhumi              | 16 / 6 / 2009   | PNG-KL      | Malindo    | 93412        |
| Ahmad Labu           | 2 / 23 / 2005   | KUL-POL     | MasAirline | 030423030779 |
| Danish               | 31 / 12 / 2030  | KUL-JHB     | MasAirline | 030423030779 |

```
void displayQueueBookings(const BookingQueue &queue) const
{
    BookingQueue::Node *current = queue.getFrontNode(); // Use the public method to get the front node

    while (current != nullptr)
    {
        current->data.displayBookingInfo(); // Display the booking info
        current = current->next;           // Move to the next node
    }
}
```

# OTHERS: APPROVE BOOKING (COMBINATION ENQUEUE AND DEQUEUE)

```
void approveReservation()
{
    if (!reservationRequests.isEmpty())
    {
        displayRequestedBookings();

        cout << "\n\t\t\t\tDo you want to approve all reservations? (1 for Yes, 0 for No): ";
        int decision;
        cin >> decision;

        if (decision == 1)
        {
            while (!reservationRequests.isEmpty())
            {
                Cust booking = reservationRequests.getFront();
                reservationRequests.dequeue();
                doneBooking.enqueue(booking);
            }
            cout << "\n\t\t\t\tAll reservations approved." << endl;
        }
        else
        {
            cout << "\n\t\t\t\tReservations not approved." << endl;
        }
    }
    else
    {
        cout << "\n\t\t\t\tNo reservation requests to approve." << endl;
    }
}
```

| Name | Date of Booking | Destination | Airlines | IC Number |
|------|-----------------|-------------|----------|-----------|
| q    | 1 / 1 / 1       | q           | q        | 1         |

Enter the number of reservations to approve (0 to cancel): 0

Reservations not approved.

**THANK YOU !**