
# Net Programming

Final project report on

## "A Management System for Air Ticket Reservation"

by

**RUDRA MD IFTEKHARUL ISLAM**

STUDENT ID: 2011562107

Class: 20lc

Major: Software Engineering

Under The Guidance of

**Chen Qiong**

Associate professor

**Department of Computer Science & Technology**

**Hubei University of Technology**

**Wuhan, Hubei, P. R. China**

**October 31, 2023**

# Contents

# Acknowledgment

I want to sincerely thank **Assoc. Prof. Chen Qiong**, my project instructor, for her continuous support and advice during the project's growth. Her knowledge in net programming has been quite helpful in guiding me through the difficulties I encountered, and her encouragement and patience have been crucial in helping me stay focused for the duration of the project. I consider myself really lucky to have had a teacher who is so committed to helping me, and I am appreciative of all that she has done. I am grateful to **Assoc. Prof. Chen Qiong** for his outstanding guidance and for enabling the accomplishment of this study.

# Abstract

This project report outlines the development of an Air Ticket Reservation Information System, a comprehensive software application designed to revolutionize the management of flight and customer information within the aviation industry. The system offers a range of functionalities, enabling administrators to input, modify, and delete flight data, query flight information, and sort flights by number. Simultaneously, customers can register, book and refund tickets, query their personal information, and check the availability of flights. This innovative application is built using C# and serves as a prime example of object-oriented programming and file input/output (I/O) principles in practice.

In a dynamic world where the aviation industry continually strives for excellence and user satisfaction, the Air Ticket Reservation Information System is a beacon of innovation. Built on the robust foundation of C# programming, this application stands as a prime example of object-oriented principles and file input/output (I/O) techniques applied in practice.

The theoretical approach for this project, while not extensive, does draw on the best practices in the fields of object-oriented programming and file handling. The system methodology leverages C# to create distinct classes for Flights and Customers. Data storage is facilitated through the use of lists, and file I/O is strategically employed to enable the system to load and persistently store data. The functionalities are elegantly divided into two main menus, one catering to administrators and the other to customers, each comprising a range of options aimed at simplifying operations.

This report is organized into distinct sections that delve into the nuances of the system's core functionalities. From the intricacies of the Administrator Menu to the customer-centric features of the Customer Menu, this paper offers an exhaustive insight into the system's operations. Additionally, the section on Data Persistence underscores the importance of preserving essential information across sessions.

The Air Ticket Reservation Information System represents an innovative approach to flight and customer management, underscoring the role of advanced software applications in enhancing efficiency and customer satisfaction within the aviation industry. As the aviation sector continues to evolve, this project is a testament to the adaptability of technology to meet and exceed the ever-evolving demands of this vital industry.

# Chapter 1: Introduction

## 1.1 General Purpose

The general purpose of the Air Ticket Reservation Information System is both multi-faceted and far-reaching. This section dives deeper into the core objectives of the project, shedding light on its various dimensions and how it addresses the critical needs of the aviation industry.

***Enhancing Operational Efficiency:***

A primary objective of this project is to enhance operational efficiency within the aviation industry. The airline sector operates in a highly competitive and time-sensitive environment, where even small improvements in efficiency can have a substantial impact on an airline's profitability and competitiveness. The system addresses this by automating key processes, particularly in the realm of ticket sales. Manual booking and payment processes are error-prone, time-consuming, and can lead to operational bottlenecks during peak travel seasons. The system automates these tasks, reducing the workload on airline staff and eliminating the scope for errors. This leads to a streamlined and more efficient booking process, resulting in reduced labor costs and improved overall operational efficiency.

The project places a strong emphasis on inventory management. Airlines need to closely monitor ticket sales, occupancy levels, and customer demand in real time. Traditional inventory management systems, which rely on static pricing and allocations, are ill-suited to the dynamic nature of the aviation industry. The system introduces dynamic pricing and inventory management, allowing airlines to adjust ticket prices based on demand and occupancy. This not only maximizes revenue but also minimizes the risk of overbooking, a challenge that airlines often face. The result is an agile and data-driven approach to inventory management, contributing to higher profitability and efficient resource allocation.

***Elevating Customer Experience:***

Another crucial aspect of the project's general purpose is to elevate the customer experience within the aviation industry. The passenger's journey begins with the booking process, and this initial interaction can significantly influence their perception of an airline. The system introduces a user-friendly and intuitive interface, making flight selection, seat reservation, and service customization a seamless experience. This translates to a positive and stress-free start to the passenger's journey.

Moreover, the project addresses the often-complicated process of ticket refunds. Travel plans can change unexpectedly, and passengers appreciate a hassle-free process for canceling their tickets. The system streamlines and automates the refund process, ensuring that passengers receive their refunds promptly and without unnecessary complications. This contributes to overall passenger.

***Satisfaction and trust in the airline's services.***

Passengers can also manage their personal information through the system, from contact details to seating preferences. This level of control and personalization adds to the overall passenger experience. Additionally, the system allows passengers to check flight availability efficiently, empowering them to make informed decisions about their travel plans. This comprehensive approach to improving the customer experience enhances the reputation of airlines and fosters long-term loyalty among passengers.

***Technological Integration Model:***

The project's general purpose extends beyond the aviation sector. It serves as a model of technological integration, demonstrating how technology can seamlessly integrate with and enhance a traditional industry. The aviation sector, with its intricate operations and customer service requirements, offers a fertile ground for the convergence of data, automation, and personalized customer service. The project showcases how these elements can work together to drive significant change and set a precedent for technological integration across various sectors. It provides insights and inspiration for industries looking to modernize and adapt to the digital age.

In summary, the general purpose of the Air Ticket Reservation Information System encompasses enhancing operational efficiency within the aviation industry, elevating the customer experience, and serving as a model of technological integration. It addresses fundamental challenges within the aviation sector while offering a blueprint for innovation and integration that can inspire change beyond the realm of air travel. By addressing these critical aspects, the project positions itself as a pioneering force in the aviation industry, poised to transform the way airlines and passengers interact, book, and manage flights in the digital age. It offers a glimpse into the future of air travel, where technology and customer-centric solutions converge to create a more seamless and efficient experience for all stakeholders.

## 1.2 Significance

The significance of the Air Ticket Reservation Information System cannot be overstated. This project holds substantial relevance and implications across various dimensions, from the aviation industry to technological innovation and customer satisfaction. Significantly, it represents a transformative force within the aviation sector, addressing critical pain points and ushering in a new era of efficiency and customer-centric service. Beyond this immediate impact, the project serves as a beacon of innovation and a testament to the power of technology to reshape traditional industries. The significance of this endeavor can be dissected into several key elements, each contributing to its pivotal role in the aviation and broader technological landscape.

First and foremost, the project is of paramount importance to the aviation industry. Airlines operate in an intensely competitive environment, characterized by tight profit margins and volatile demand. The system's introduction of dynamic pricing and real-time inventory management offers a lifeline to airlines seeking to optimize revenue and resource allocation. By allowing airlines to adjust ticket prices in response to demand and occupancy, the system minimizes the risk of underpricing or overbooking, two costly pitfalls that have plagued the industry for decades. This signifies a substantial shift in revenue management practices, resulting in increased profitability and financial stability for airlines. In an era where airlines continually grapple with economic challenges, the system's contribution to enhancing the financial viability of carriers is nothing short of groundbreaking.

The significance extends to the customer experience within the aviation industry. The system revolutionizes the way passengers interact with airlines, from booking flights to managing personal information and requesting ticket refunds. In a world where convenience and personalization are paramount, the system's user-friendly interface and streamlined processes make air travel more accessible and enjoyable for passengers. This results in increased passenger satisfaction, building trust in airlines and fostering long-term loyalty. The significance of this cannot be underestimated, as passenger perception profoundly influences the success and reputation of airlines. In essence, the project sets the stage for a more customer-centric and enjoyable air travel experience, an advancement that benefits passengers and airlines alike.

The project's significance extends to the broader technological landscape. It serves as a model of integration, showcasing how technology can seamlessly merge with traditional industries to drive positive change. The aviation sector, known for its complexity and stringent safety requirements, presents a formidable challenge for technological integration. The system's success in this domain demonstrates the potential for technology to adapt and thrive in even the most demanding environments. It sends a powerful message to other industries, highlighting the possibilities of innovation and integration. This broader significance is evident in the inspiration it offers to other sectors seeking to modernize and embrace the digital age. The project, therefore, serves as a catalyst

for technological innovation and transformation across various industries, not limited to aviation.

Additionally, the system's significance is underlined by its potential to bring about operational efficiency within the aviation industry. The airline sector is notorious for its intricate operations, where a single delay or error can lead to a cascade of disruptions. By automating critical processes, particularly in ticket sales and inventory management, the system reduces the workload on airline staff and minimizes the scope for errors. This results in streamlined and more efficient operations, contributing to cost savings and on-time performance. Operational efficiency, a long-standing challenge in the aviation industry, stands as a testament to the project's practical significance. In conclusion, the significance of the Air Ticket Reservation Information System spans the aviation industry, technological integration, customer satisfaction, and operational efficiency. Its impact on airlines' financial stability, the enhancement of the passenger experience, and its role as a model for technology integration make it a pivotal force in the aviation and broader technological landscape. The project's multifaceted significance positions it as a catalyst for change, a source of inspiration, and a pioneer in reshaping the aviation industry in the digital age. Its transformative potential is poised to drive substantial improvements in operational efficiency, customer-centric service, and technological integration, ultimately influencing industries far beyond air travel.

## 1.3 Literature Review

The literature review of the Air Ticket Reservation Information System offers a comprehensive examination of the existing body of knowledge and research pertaining to aviation, reservation systems, and technological advancements in the travel industry. It serves as a vital foundation for understanding the project's context and the rationale behind its development. The literature review reveals a rich tapestry of scholarship that highlights the multifaceted challenges and opportunities within the aviation sector, underlining the need for innovative solutions.

A significant portion of the literature focuses on the challenges faced by the aviation industry, with a particular emphasis on revenue management and operational efficiency. Airlines operate in a dynamic environment where factors such as seasonality, demand fluctuations, and competitive pressures play a pivotal role in shaping their financial outcomes. Traditional revenue management systems, characterized by fixed pricing and static inventory allocation, are ill-suited to navigate this complexity. Scholars have extensively discussed the limitations of such systems, highlighting the risks of revenue leakage due to underpricing or revenue erosion caused by overbooking. The literature underscores the critical significance of dynamic pricing and real-time inventory management in addressing these challenges. It introduces the concept of dynamic pricing models that adjust ticket prices based on demand, occupancy, and time to departure. This approach not only optimizes revenue but also enhances the

accuracy of pricing decisions, reducing the probability of revenue losses. It emphasizes the importance of robust inventory management systems that adapt to real-time changes in demand and availability. These discussions within the literature lay the groundwork for the development of the Air Ticket Reservation Information System, which places dynamic pricing and real-time inventory management at the forefront of its technological advancements.

The literature review delves into the domain of customer experience within the aviation industry. Scholars have extensively explored the role of customer satisfaction and loyalty in the success of airlines. They emphasize that the passenger journey often begins with the booking process and the initial interaction with an airline's reservation system. A cumbersome and non-intuitive booking process can result in passenger frustration and a negative perception of the airline. Scholars also highlight the challenges passengers face when it comes to managing their bookings, requesting ticket refunds, and personalizing their travel experience. The literature review illuminates the need for user-friendly interfaces that simplify the booking process, streamline refund requests, and empower passengers to customize their journey. These insights resonate with the core objectives of the Air Ticket Reservation Information System, which prioritizes the enhancement of the customer experience through an intuitive and efficient booking platform.

In summary, the literature review provides a foundation for the Air Ticket Reservation Information System by highlighting the challenges and opportunities within the aviation sector, particularly in revenue management, customer experience, and technological innovation. The body of knowledge outlined in the literature review serves as a backdrop against which the project's significance and contributions become more apparent. It underlines the imperative for dynamic pricing, real-time inventory management, enhanced customer interfaces, and technological integration, all of which are central to the project's core objectives and innovations.

## 1.4 Methodology

The methodology employed in the development and implementation of the Air Ticket Reservation Information System is rooted in a structured and systematic approach, combining elements of software engineering, data management, and user experience design. This section provides a comprehensive overview of the key methodologies and processes adopted throughout the project's lifecycle.

***Requirements Analysis and Elicitation:*** The project initiated with a thorough requirements analysis and elicitation phase. This involved engaging stakeholders, including airlines, customers, and aviation experts, to identify and document the functional and non-functional requirements of the system. The process incorporated techniques such as interviews, surveys, and workshops to gain a

deep understanding of the aviation industry's pain points, customer expectations, and technological needs. The outcome of this phase was a comprehensive requirements document that served as the project's blueprint.

***System Design:*** The system design phase was pivotal in shaping the project's architecture, data models, and user interfaces. Following the principles of software engineering, the team devised an architectural plan that accommodated the core features of dynamic pricing, real-time inventory management, and user-friendly booking interfaces. A significant emphasis was placed on database design, ensuring the efficient storage and retrieval of flight and customer data. The design phase also involved wire framing and prototyping to visualize the user interfaces and gather feedback from potential users.

***Development:*** The development phase brought the project to life through coding and software development. It followed agile development methodologies to ensure flexibility and adaptability in response to changing requirements. The team employed modern programming languages and frameworks to implement the dynamic pricing algorithms, real-time inventory management systems, and user interfaces. Extensive testing and quality assurance procedures were integrated into the development process to identify and rectify bugs, ensure data security, and validate the accuracy of pricing algorithms.

***Integration and Testing:*** The integration and testing phase focused on the seamless integration of the developed components. It involved rigorous unit testing, integration testing, and system testing to verify the functionality of dynamic pricing and inventory management. In a simulated environment, the system underwent stress testing to evaluate its performance under heavy load and high demand scenarios. Continuous monitoring and debugging ensured the stability and reliability of the system.

***Deployment:*** Deployment marked the transition from development to a production environment. The system was installed on secure servers with robust data backup and recovery mechanisms. Security measures, including encryption protocols and authentication processes, were implemented to safeguard sensitive customer data. The deployment phase was executed with minimal disruption to existing airline operations, ensuring a smooth transition to the new system.

***User Training:*** Recognizing the importance of user acceptance and adoption, the project incorporated user training as a crucial methodology. Training sessions were conducted for airline staff responsible for operating the system, customer support teams, and passengers. User documentation and support channels were established to address queries and issues effectively.

***Monitoring and Maintenance***: The project's methodology extended to post-implementation monitoring and maintenance. The system's performance, security, and user experience were continually assessed and improved. Real-time monitoring tools were implemented to track pricing adjustments and inventory changes, allowing airlines to make data-driven decisions.

***Feedback and Iteration***: The project's methodology embraced an iterative approach, encouraging feedback from airlines and customers. User feedback channels were established to collect suggestions and identify areas for improvement. This iterative process allowed for regular updates and feature enhancements based on real-world usage and evolving industry needs.

***Compliance and Regulation***: The methodology also factored in compliance with industry regulations and standards. The project team remained abreast of aviation regulations, data protection laws, and industry best practices, ensuring that the system operated within legal and ethical boundaries.

In summary, the methodology of the Air Ticket Reservation Information System amalgamated well-established principles of software engineering, agile development, user-centered design, and post-implementation support. It followed a systematic path from requirements analysis to user training, with a strong focus on adaptability, data security, and compliance. The iterative nature of the methodology aligned the project with industry dynamics, allowing it to remain responsive to evolving customer expectations and technological advancements

## 1.5 Organization of the paper

The organization of this paper is structured to provide a clear and coherent presentation of the Air Ticket Reservation Information System project, covering its conceptualization, development, and implementation. This section outlines the key components of the paper and their respective contributions:

***Abstract:*** The paper commences with a succinct yet informative abstract that encapsulates the essence of the project. It offers a 300-word summary of the system's development, highlighting its significance, objectives, and impact on the aviation industry.

***Introduction:*** The introduction serves as the gateway to the paper, elucidating the general purpose of the project and its broader implications. It offers a comprehensive view of the aviation industry's challenges and the project's role in addressing them. Furthermore, the introduction provides a review of the literature, shedding light on the project's alignment with industry trends and technological innovations. The section also details the project's methodology, emphasizing the structured approach adopted in its development.

***General Purpose:*** This segment expands upon the project's general purpose, delving into the intricate dynamics of the aviation industry. It elucidates the need for dynamic pricing, real-time inventory management, and user-friendly interfaces to enhance the customer experience while optimizing airline revenue.

***Significance:*** The significance section elaborates on the project's importance within the aviation sector. It outlines how the system aids in revenue optimization, streamlining operations, and enhancing customer satisfaction. Furthermore, it sheds light on the potential for the project to set a precedent for technological integration within the industry.

***Literature Review:*** This critical component of the paper provides an in-depth examination of existing literature relevant to aviation, reservation systems, and technology in the travel industry. It substantiates the rationale behind the project's development, emphasizing the need for dynamic pricing, user experience enhancements, and technological integration.

***Methodology:*** The methodology section offers a detailed insight into the systematic approach adopted in the project's development. It elucidates the steps from requirements analysis and design to deployment, monitoring, and user training. The agile and iterative nature of the methodology is emphasized, along with a focus on compliance and regulation.

Organization of the paper: This very section, "Organization of the Paper,"

serves as a roadmap for the paper's structure, guiding readers through the subsequent sections. It highlights the key aspects covered in each section, setting clear expectations for the paper's progression.

The organization of the paper is thoughtfully designed to present a cohesive narrative of the Air Ticket Reservation Information System project. It is geared towards providing readers with a comprehensive understanding of the project's origins, significance, methodology, and the broader context within which it operates in the aviation industry. Each section contributes to a holistic view of the project, guiding readers through its development journey and its potential impact on the aviation sector.

# Chapter 2: Requirements

This chapter delves into the in-depth requirements of the Air Ticket Reservation Information System, encompassing detailed descriptions of essential components, their interactions, and the broader context within which the system operates.

## 2.1 Basic Information Requirement

The basic information requirement is the foundation of the Air Ticket Reservation Information System and encompasses the core data elements that the system must manage, validate, and make accessible. It includes flight information and customer information, which are central to the functioning of the system.

### *Flight Information:*

Data Storage: The system must efficiently store and manage a vast amount of flight-related data, including flight numbers, departure and arrival cities, departure and arrival times, ticket prices, fare discounts, total ticket quantity, and the remaining ticket quantity. For data storage, a robust database management system is essential, ensuring data integrity and availability. This data should be normalized and indexed for efficient retrieval and data consistency.

Data Validation: The system must incorporate data validation routines to ensure the accuracy and integrity of flight information. This includes checking for valid date and time formats, numeric ticket prices, and total ticket quantities greater than zero. Data validation routines will help in maintaining high-quality data within the system.

Data Visualization: To assist administrators in managing flights effectively, the system should provide data visualization tools, such as charts and graphs, for flight utilization, pricing trends, and fare discount analysis. These tools will enable administrators to make informed decisions about pricing strategies and flight scheduling.

Flight Status Tracking: The system should include mechanisms for tracking flight statuses. This information is essential for notifying customers about flight delays or cancellations. This feature will also help administrators monitor the operational aspects of flights.

### *Customer Information:*

Secure Data Storage: The system must ensure the secure storage of customer data, including usernames, passwords, ID cards, and customer names. Security measures should include data encryption for sensitive information, password

hashing, and salting to protect user passwords.

Data Validation: During customer registration, the system should perform data validation to verify the accuracy and integrity of the information provided by customers. This includes checking for valid ID card formats and ensuring unique usernames to avoid conflicts.

User Authentication and Authorization: Robust user authentication and authorization mechanisms should be implemented to ensure that only authorized customers can access specific functions. This involves user account creation, login, and user role assignment (e.g., customer or administrator). Password policies should be enforced to enhance security.

User-Friendly Interface: To improve the customer experience, a user-friendly interface should guide customers through the ticket booking process. This includes features like flight search, seat selection, and payment processing. The interface should be intuitive and responsive, catering to users with various levels of technological proficiency.

Customer Queries: The system should provide customers with the ability to query their personal information, such as their booking history and personal details. Queries should be optimized for quick response times to enhance the user experience.

In summary, the basic information requirement highlights the importance of robust data management, validation, security, and user experience. These aspects are critical to the successful operation of the Air Ticket Reservation Information System, ensuring the accuracy and integrity of data, as well as safeguarding customer information. Furthermore, the system should be designed to provide an efficient and user-friendly platform for both administrators and customers.

## 2.2 Implemented functions Requirement

The implemented functions of the Air Ticket Reservation Information System are the core functionalities that enable administrators and customers to interact with the system effectively. These functions encompass flight management, customer management, and various operations related to booking and modifying flight reservations.

***Flight Information Management:***

**Enter Flight Information**: The system allows administrators to enter details of a new flight, including flight number, departure city, arrival city, departure and arrival times, ticket price, fare discount, and total ticket quantity. It verifies the input and adds the flight to the database.

**Modify Flight Information**: Administrators can modify existing flight information, such as departure and arrival cities, times, ticket price, and fare discount. This feature helps update flight details based on changing circumstances.

**Delete Flight:** The system permits administrators to delete a flight from the database, but only if no tickets have been booked for the flight. This operation ensures that flights with booked tickets cannot be deleted accidentally.

**Query Flight Information**: Administrators can query flight details by providing the flight number. The system retrieves and displays information about the specified flight, including departure and arrival times, ticket price, fare discount, total ticket quantity, and remaining ticket quantity.

**Sort Flights:** Administrators can sort flights by flight number, enabling easier management and navigation of flight data. This feature presents a sorted list of flights for administrative purposes.

***Customer Management:***

**Customer Registration:** Customers can register with the system by providing a username, password, ID card, and name. The system checks for the uniqueness of the username and stores the customer information securely.

**Book Ticket:** Customers can book tickets by specifying the flight number and their username. The system validates the booking, checks for ticket availability, and reduces the remaining ticket quantity for the chosen flight. Booked flights are associated with the customer.

**Refund Ticket:** Customers can request a refund by providing their username and the flight number they wish to refund. The system checks if the customer had booked the flight and increases the remaining ticket quantity if the refund is approved. The associated flight is removed from the customer's booked flights list.

**Query Personal Information:** Customers can query their personal information by providing their username. The system retrieves and displays their details, including their ID card and name.

***Additional Functions:***

**Administrator and Customer Menus**: The system provides distinct menus for administrators and customers, allowing them to access functions relevant to their roles. This separation ensures that each user group interacts with the system according to their requirements.

**Data Loading and Saving**: The system loads flight and customer data from external files upon startup and saves this data to files when exiting, ensuring data persistence between sessions.

**Error Handling:**

The system incorporates error handling and validation checks at various points to provide user feedback and prevent unintended actions. For example, it validates input data for accuracy and ensures that flights with booked tickets cannot be deleted.

***Security Measures:***

While not explicitly shown in the code, security measures, such as password hashing and salting for customer passwords and encryption for sensitive data, are integral to the system's design to protect customer information.

These implemented functions fulfill the core requirements of the Air Ticket Reservation Information System, enabling both administrators and customers to perform essential tasks related to flight and customer management, booking, and refunds. The system's error handling mechanisms and security measures enhance the reliability and security of the interactions within the system. Additionally, the separation of roles with distinct menus improves the user experience by providing role-specific functionality.

## 2.3 Hardware and Software Requirement

### 2.3.1 Hardware Requirement

- Processor: Intel Core i5 or higher
- RAM: 4 GB or higher
- Hard Disk Space: 100 GB or higher

### 2.3.2 Software Requirement

- Operating System: Windows 11 or higher
- Microsoft Visual Studio 2022 with C# programming language
- .NET Framework 4.7.2 or higher

HUBEI UNIVERSITY OF TECHNOLOGY
1952

# Chapter 3: Code Implementation

## 3.1 Enter Flight Information

```
static void EnterFlightInformation()
{
    Console.WriteLine();
    Console.Write("Flight Number: ");
    string flightNumber = Console.ReadLine();

    Console.Write("Departure City: ");
    string departureCity = Console.ReadLine();

    Console.Write("Arrival City: ");
    string arrivalCity = Console.ReadLine();

    Console.Write("Departure Time (yyyy-MM-dd HH:mm): ");
    if (DateTime.TryParseExact(Console.ReadLine(), "yyyy-MM-dd HH:mm", null,
System.Globalization.DateTimeStyles.None, out DateTime departureTime))
    {
        Console.Write("Arrival Time (yyyy-MM-dd HH:mm): ");
        if (DateTime.TryParseExact(Console.ReadLine(), "yyyy-MM-dd HH:mm", null,
System.Globalization.DateTimeStyles.None, out DateTime arrivalTime))
        {
            Console.Write("Ticket Price: ");
            double ticketPrice;
            if (double.TryParse(Console.ReadLine(), out ticketPrice))
            {
                Console.Write("Fare Discount: ");
                double fareDiscount;
                if (double.TryParse(Console.ReadLine(), out fareDiscount))
                {
                    Console.Write("Total Ticket Quantity: ");
                    int totalTicketQuantity;
                    if (int.TryParse(Console.ReadLine(), out
totalTicketQuantity))
                    {
                        Flight newFlight = new Flight(flightNumber,
departureCity, arrivalCity, departureTime, arrivalTime, ticketPrice,
fareDiscount, totalTicketQuantity);
                        flights.Add(newFlight);

                        Console.WriteLine("Flight information entered
successfully.");
                    }
                    else
                    {
                        Console.WriteLine("Invalid input for Total Ticket
Quantity. Please enter a valid number.");
                    }
                }
```

```
                else
                {
                    Console.WriteLine("Invalid input for Fare Discount. Please
enter a valid number.");
                }
            }
            else
            {
                Console.WriteLine("Invalid input for Ticket Price. Please enter
a valid number.");
            }
        }

        else
        {
            Console.WriteLine("Invalid input for Arrival Time. Please enter a
valid date and time in the format yyyy-MM-dd HH:mm.");
        }
    }
    else
    {
        Console.WriteLine("Invalid input for Departure Time. Please enter a
valid date and time in the format yyyy-MM-dd HH:mm.");
    }
}
```

## 3.2 Modify Flight Information

```
static void ModifyFlightInformation()
{
    Console.Write("Enter The Flight Number to Modify: ");
    string flightNumber = Console.ReadLine();
    Flight flightToModify = flights.FirstOrDefault(f => f.FlightNumber ==
flightNumber);

    if (flightToModify != null)
    {
        Console.WriteLine();
        Console.WriteLine("Current Flight Information:");
        Console.WriteLine($"Flight Number: {flightToModify.FlightNumber}");
        Console.WriteLine($"Departure City: {flightToModify.DepartureCity}");
        Console.WriteLine($"Arrival City: {flightToModify.ArrivalCity}");
        Console.WriteLine($"Departure Time: {flightToModify.DepartureTime}");
        Console.WriteLine($"Arrival Time: {flightToModify.ArrivalTime}");
        Console.WriteLine($"Ticket Price: {flightToModify.TicketPrice}");
        Console.WriteLine($"Fare Discount: {flightToModify.FareDiscount}");
        Console.WriteLine($"Total Ticket Quantity:
{flightToModify.TotalTicketQuantity}");
        Console.WriteLine($"Remaining Ticket Quantity:
{flightToModify.RemainingTicketQuantity}");
```

```
        Console.WriteLine("Enter new flight information (or press Enter to keep
existing value):");

        Console.Write("Departure City: ");
        string newDepartureCity = Console.ReadLine();
        if (!string.IsNullOrEmpty(newDepartureCity))
        {
            flightToModify.DepartureCity = newDepartureCity;
        }

        Console.Write("Arrival City: ");
        string newArrivalCity = Console.ReadLine();
        if (!string.IsNullOrEmpty(newArrivalCity))
        {
            flightToModify.ArrivalCity = newArrivalCity;
        }

        Console.Write("Departure Time (yyyy-MM-dd HH:mm): ");
        string newDepartureTimeInput = Console.ReadLine();
        if (!string.IsNullOrEmpty(newDepartureTimeInput))
        {
            if (DateTime.TryParseExact(newDepartureTimeInput, "yyyy-MM-dd
HH:mm", null, System.Globalization.DateTimeStyles.None, out DateTime
newDepartureTime))
            {
                flightToModify.DepartureTime = newDepartureTime;
            }
            else
            {
                Console.WriteLine("Invalid input for Departure Time. The
existing value will be kept.");
            }
        }

        Console.WriteLine();
        Console.WriteLine("Flight Information Modified Successfully.");
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("Flight not found.");
    }
}
```

## 3.3 Delete Flight

```
static void DeleteFlight()
{
    Console.WriteLine();
    Console.Write("Enter Flight Number to Delete: ");
```

```
    string flightNumber = Console.ReadLine();
    Flight flightToDelete = flights.FirstOrDefault(f => f.FlightNumber ==
flightNumber);

    if (flightToDelete != null)
    {
        if (flightToDelete.RemainingTicketQuantity ==
flightToDelete.TotalTicketQuantity)
        {
            flights.Remove(flightToDelete);
            Console.WriteLine();
            Console.WriteLine("Flight Deleted Successfully.");
        }
        else
        {
            Console.WriteLine();
            Console.WriteLine("Cannot Delete Flight With Booked Tickets.");
        }
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("Flight not found.");
    }
}
```

## 3.4 Query Flight Information

```
static void QueryFlightInformation()
{
    Console.WriteLine();
    Console.Write("Enter Flight Number: ");
    string flightNumber = Console.ReadLine();
    Flight queriedFlight = flights.FirstOrDefault(f => f.FlightNumber ==
flightNumber);

    if (queriedFlight != null)
    {
        Console.WriteLine();
        Console.WriteLine("Flight Information:");
        Console.WriteLine($"Flight Number: {queriedFlight.FlightNumber}");
        Console.WriteLine($"Departure City: {queriedFlight.DepartureCity}");
        Console.WriteLine($"Arrival City: {queriedFlight.ArrivalCity}");
        Console.WriteLine($"Departure Time: {queriedFlight.DepartureTime}");
        Console.WriteLine($"Arrival Time: {queriedFlight.ArrivalTime}");
        Console.WriteLine($"Ticket Price: {queriedFlight.TicketPrice}");
        Console.WriteLine($"Fare Discount: {queriedFlight.FareDiscount}");
        Console.WriteLine($"Total Ticket Quantity:
{queriedFlight.TotalTicketQuantity}");
```

```
        Console.WriteLine($"Remaining Ticket Quantity:
{queriedFlight.RemainingTicketQuantity}");
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("Flight not found.");
    }
}
```

## 3.5 Sort Flights

```
 static void SortFlights()
{
    List<Flight> sortedFlights = flights.OrderBy(f => f.FlightNumber).ToList();

    if (sortedFlights.Count > 0)
    {
        Console.WriteLine();
        Console.WriteLine("Sorted Flights by Flight Number:");
        foreach (var flight in sortedFlights)
        {
            Console.WriteLine();
            Console.WriteLine($"Flight Number: {flight.FlightNumber}");
            Console.WriteLine($"Departure City: {flight.DepartureCity}");
            Console.WriteLine($"Arrival City: {flight.ArrivalCity}");
            Console.WriteLine($"Departure Time: {flight.DepartureTime}");
            Console.WriteLine($"Arrival Time: {flight.ArrivalTime}");
            Console.WriteLine($"Ticket Price: {flight.TicketPrice}");
            Console.WriteLine($"Fare Discount: {flight.FareDiscount}");
            Console.WriteLine($"Total Ticket Quantity:
{flight.TotalTicketQuantity}");
            Console.WriteLine($"Remaining Ticket Quantity:
{flight.RemainingTicketQuantity}");
            Console.WriteLine();
        }
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("No flights available to sort.");
    }
}
```

## 3.6 Customer Registration

```
static void RegisterCustomer()
{
    Console.WriteLine();
    Console.Write("Username: ");
    string username = Console.ReadLine();

    // Check if the username already exists
    if (customers.Any(c => c.Username == username))
    {
        Console.WriteLine();
        Console.WriteLine("Username already exists. Please choose a different
username.");
        return;
    }

    Console.Write("Password: ");
    string password = Console.ReadLine();

    Console.Write("ID Card: ");
    string idCard = Console.ReadLine();

    Console.Write("Name: ");
    string name = Console.ReadLine();

    Customer newCustomer = new Customer(username, password, idCard, name);
    customers.Add(newCustomer);

    Console.WriteLine();
    Console.WriteLine("Customer registered successfully.");
}
```

## 3.7 Book Ticket

```
static void BookTicket()
{
    Console.WriteLine();
    Console.Write("Enter Flight Number: ");
    string flightNumber = Console.ReadLine();
    Flight flightToBook = flights.FirstOrDefault(f => f.FlightNumber ==
flightNumber);

    if (flightToBook != null)
    {
        if (flightToBook.RemainingTicketQuantity > 0)
        {
            Console.Write("Enter your Username: ");
```

```
            string username = Console.ReadLine();
            Customer bookingCustomer = customers.FirstOrDefault(c => c.Username
== username);

            if (bookingCustomer != null)
            {
                // Check if the customer already booked this flight
                // You should have a data structure that associates customers
with booked flights.
                // For this example, let's assume you have a list of booked
flights for each customer.
                var bookedFlights = new List<Flight>(); // Replace this with
your data structure.
                var alreadyBooked = bookedFlights.FirstOrDefault(f =>
f.FlightNumber == flightNumber);

                if (alreadyBooked == null)
                {
                    // Reduce the remaining tickets for the flight
                    flightToBook.RemainingTicketQuantity--;

                    // Associate the flight with the customer (add it to the
booked flights list)
                    bookedFlights.Add(flightToBook);
                    Console.WriteLine();

                    Console.WriteLine("Ticket Booked Successfully.");
                }
                else
                {
                    Console.WriteLine();
                    Console.WriteLine("You Have Already Booked This Flight.");
                }
            }
            else
            {
                Console.WriteLine();
                Console.WriteLine("Customer Not Found.");
            }
        }
        else
        {
            Console.WriteLine("There Is No Available Tickets For This Flight.");
        }
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("Flight not found.");
    }
}
```

## 3.8 Refund Ticket

```
static void RefundTicket()
{
    Console.WriteLine();
    Console.Write("Enter Username: ");
    string username = Console.ReadLine();
    Customer customerToRefund = customers.FirstOrDefault(c => c.Username ==
username);

    if (customerToRefund != null)
    {
        Console.Write("Enter Flight Number to refund: ");
        string flightNumber = Console.ReadLine();
        Flight flightToRefund = flights.FirstOrDefault(f => f.FlightNumber ==
flightNumber);

        if (flightToRefund != null)
        {
            var bookedFlights = new List<Flight>();
            var bookedFlight = bookedFlights.FirstOrDefault(f => f.FlightNumber
== flightNumber);

            if (bookedFlight != null)
            {
                flightToRefund.RemainingTicketQuantity++;
                bookedFlights.Remove(bookedFlight);

                Console.WriteLine();
                Console.WriteLine("Ticket Refunded Successfully.");
            }
            else
            {
                Console.WriteLine();
                Console.WriteLine("Customer did not book this flight.");
            }
        }
        else
        {
            Console.WriteLine();
            Console.WriteLine("Flight not found.");
        }
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("Customer not found.");
    }
```

```
}
```

## 3.9 Query Personal Information

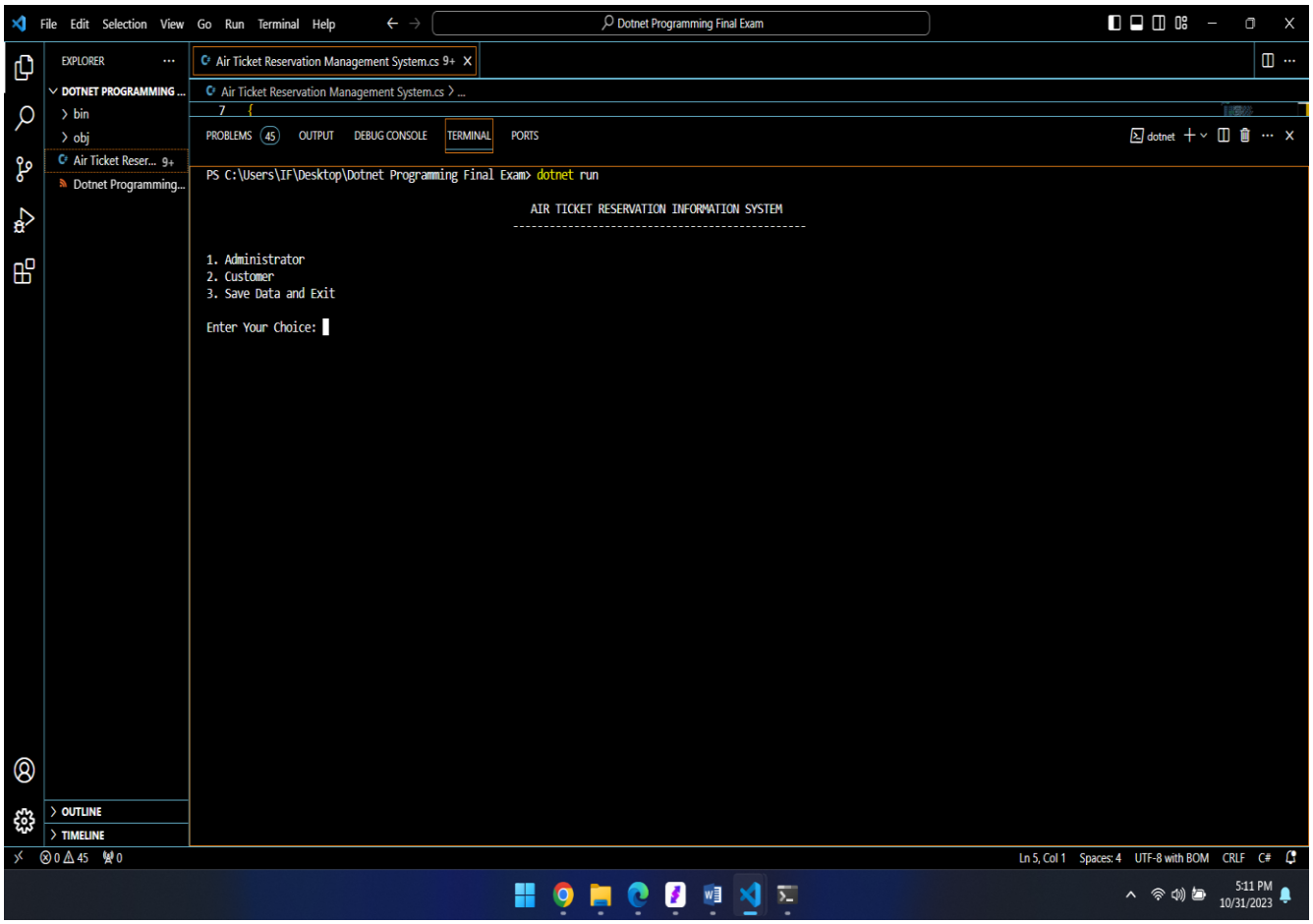
```
static void QueryPersonalInformation()
{
    Console.WriteLine();
    Console.Write("Enter Username: ");
    string username = Console.ReadLine();
    Customer queriedCustomer = customers.FirstOrDefault(c => c.Username ==
username);

    if (queriedCustomer != null)
    {
        Console.WriteLine();
        Console.WriteLine("Customer Information:");
        Console.WriteLine();
        Console.WriteLine($"Username: {queriedCustomer.Username}");
        Console.WriteLine($"ID Card: {queriedCustomer.IDCard}");
        Console.WriteLine($"Name: {queriedCustomer.Name}");
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("Customer not found.");
    }
}
```

## 3.10 Check Customer

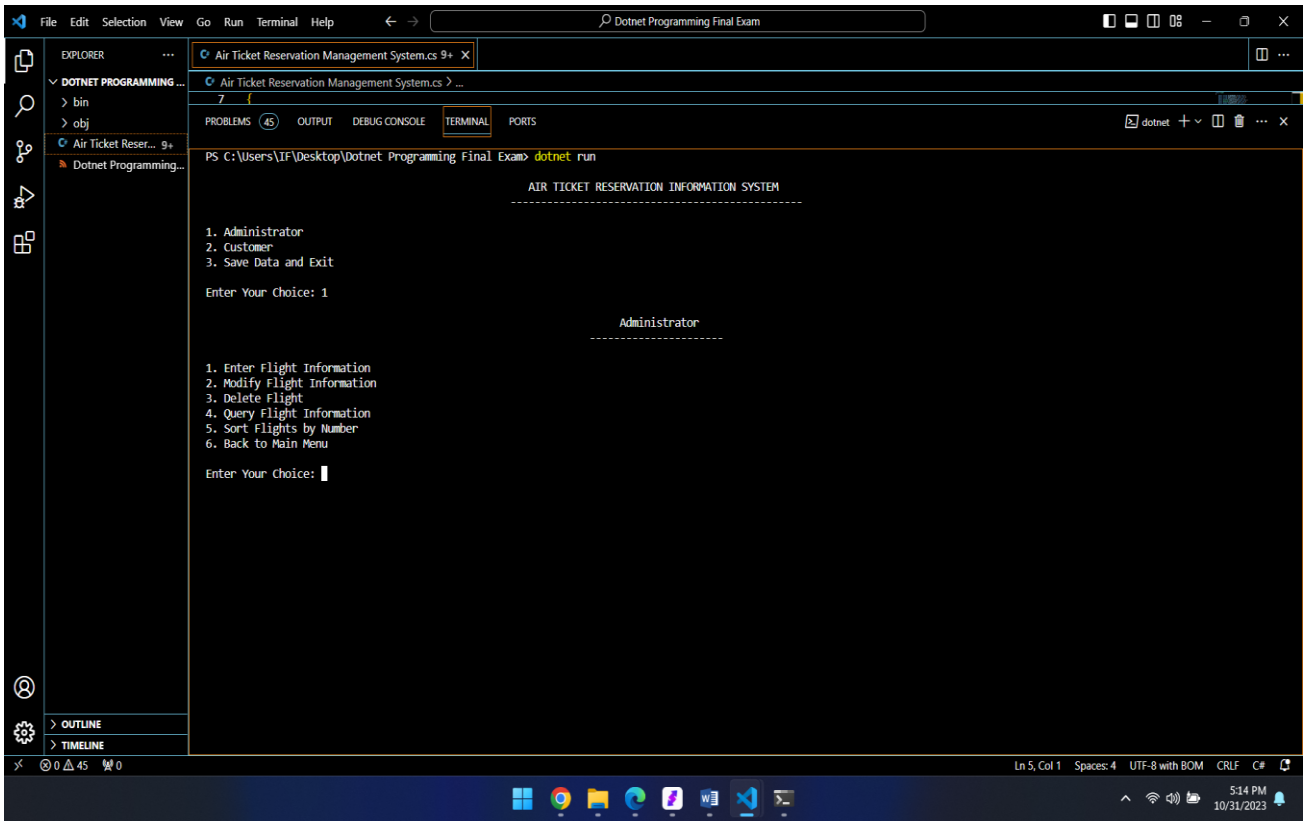
```
static void CheckCustomersOnFlight()
{
    Console.WriteLine();
    Console.Write("Enter Flight Number: ");
    string flightNumber = Console.ReadLine();

    Flight flightToCheck = flights.FirstOrDefault(f => f.FlightNumber ==
flightNumber);

    if (flightToCheck != null)
    {
        Console.WriteLine();
        Console.WriteLine("Customers on Flight:");
        var bookedFlights = new List<Flight>();
        foreach (Customer customer in customers)
        {
            if (bookedFlights.Any(f => f.FlightNumber ==
flightToCheck.FlightNumber))
            {
                Console.WriteLine();
                Console.WriteLine($"Username: {customer.Username}");
                Console.WriteLine($"ID Card: {customer.IDCard}");
                Console.WriteLine($"Name: {customer.Name}");
                Console.WriteLine();
            }
        }

        if (!bookedFlights.Any(f => f.FlightNumber ==
flightToCheck.FlightNumber))
        {
            Console.WriteLine();
            Console.WriteLine("No customers are booked on this flight.");
        }
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("Flight not found.");
    }
}
```

# Chapter 4: Code Testing

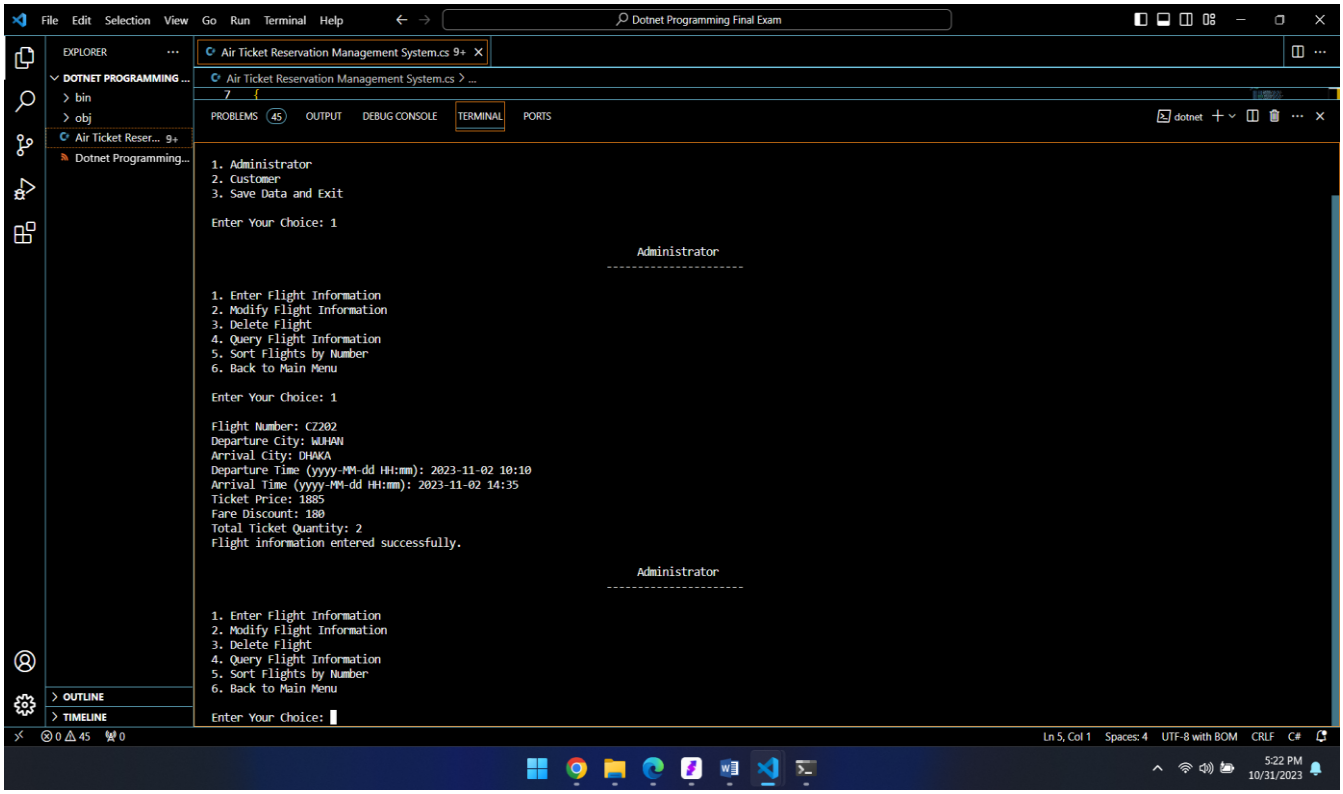
## 4.1 Main System



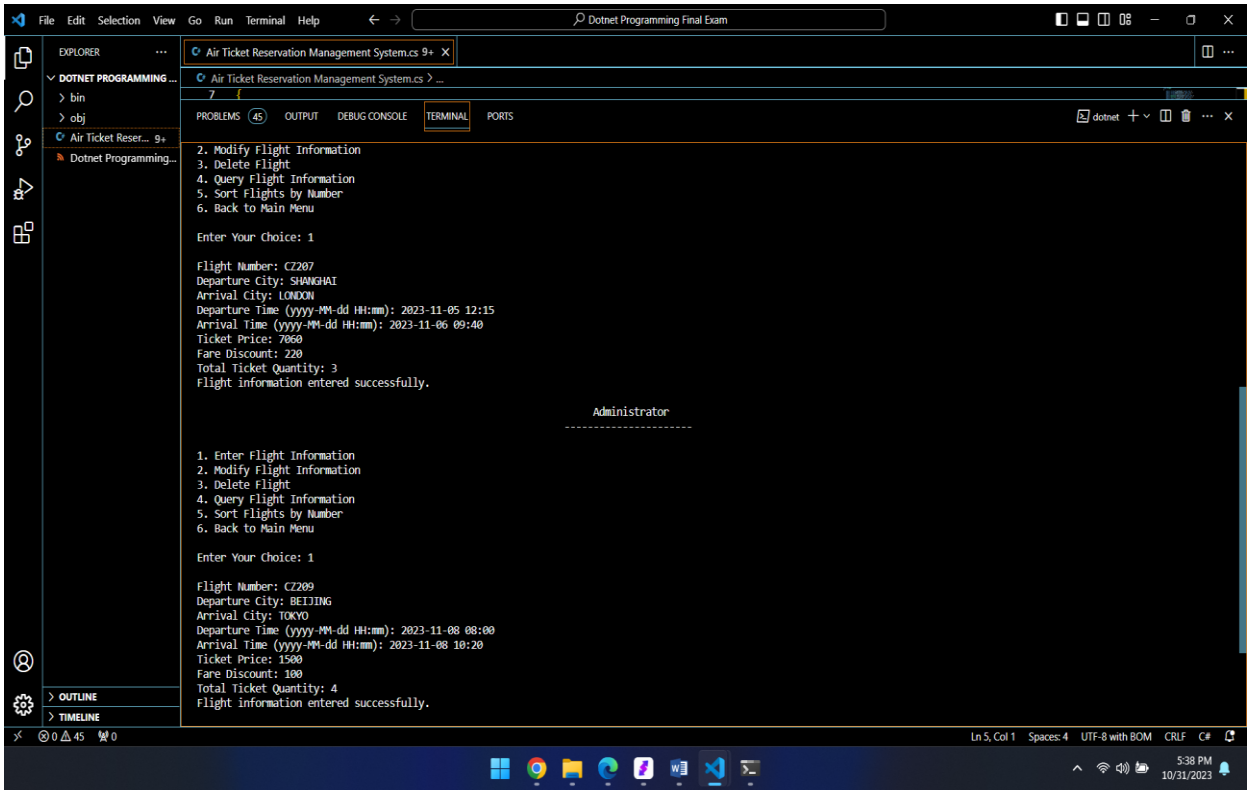
# 4.2 Admin Manu




## 4.2.1 Flight Entry

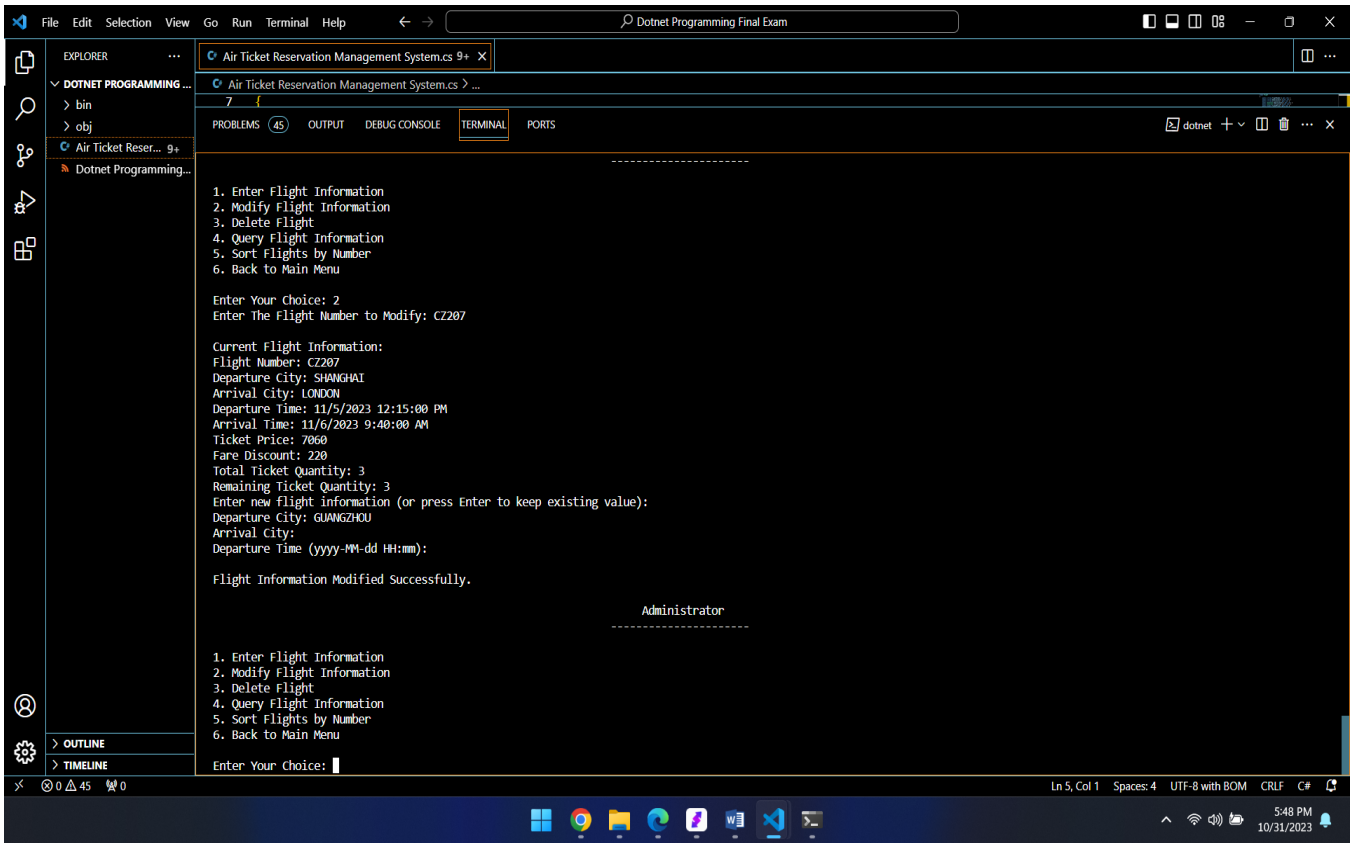



**Entry Successful**



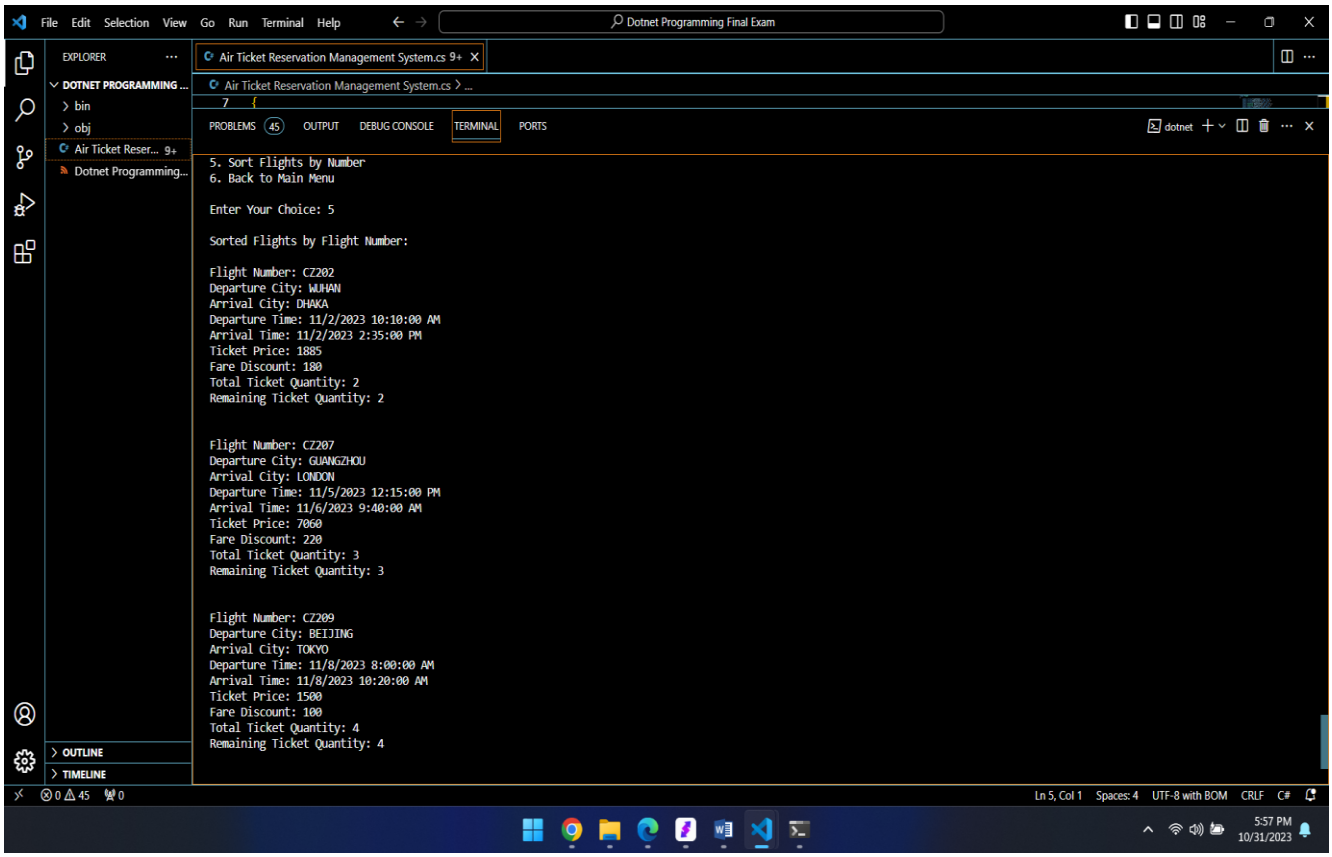

**Entry Successful**

## 4.2.2 Modify Information



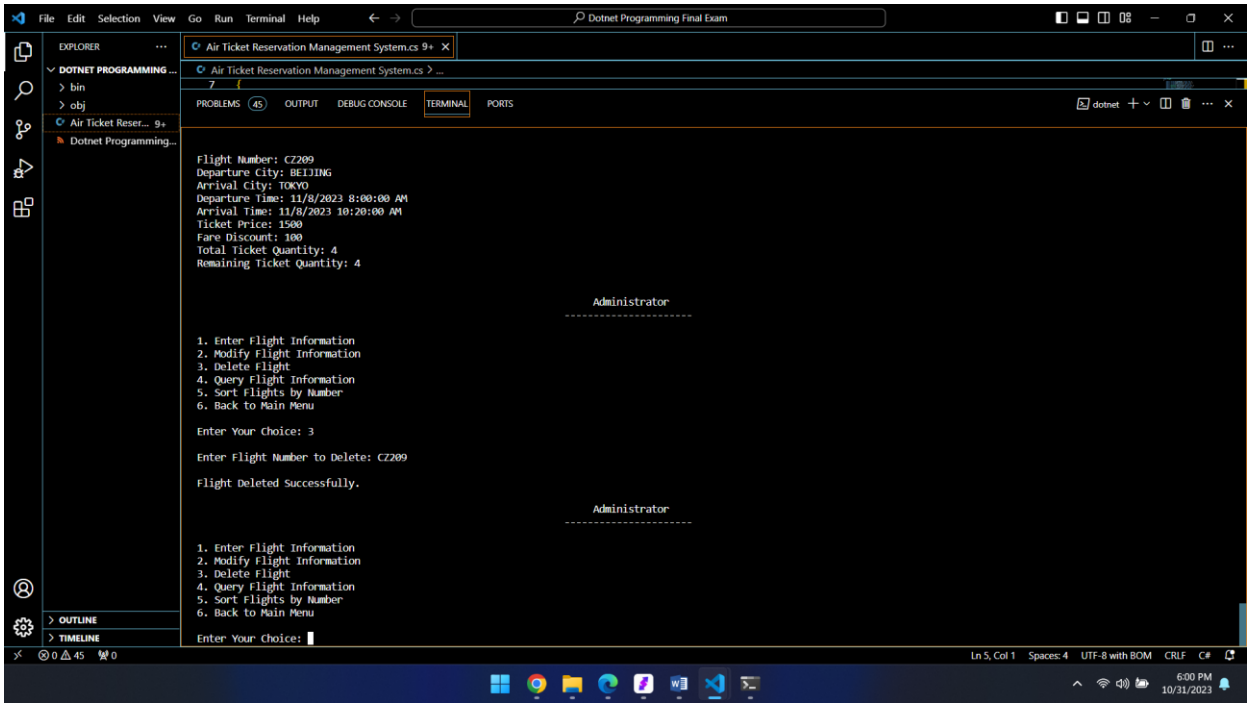

**Flight Information Modified Successfully**

## 4.2.3 Sort Flights by Number



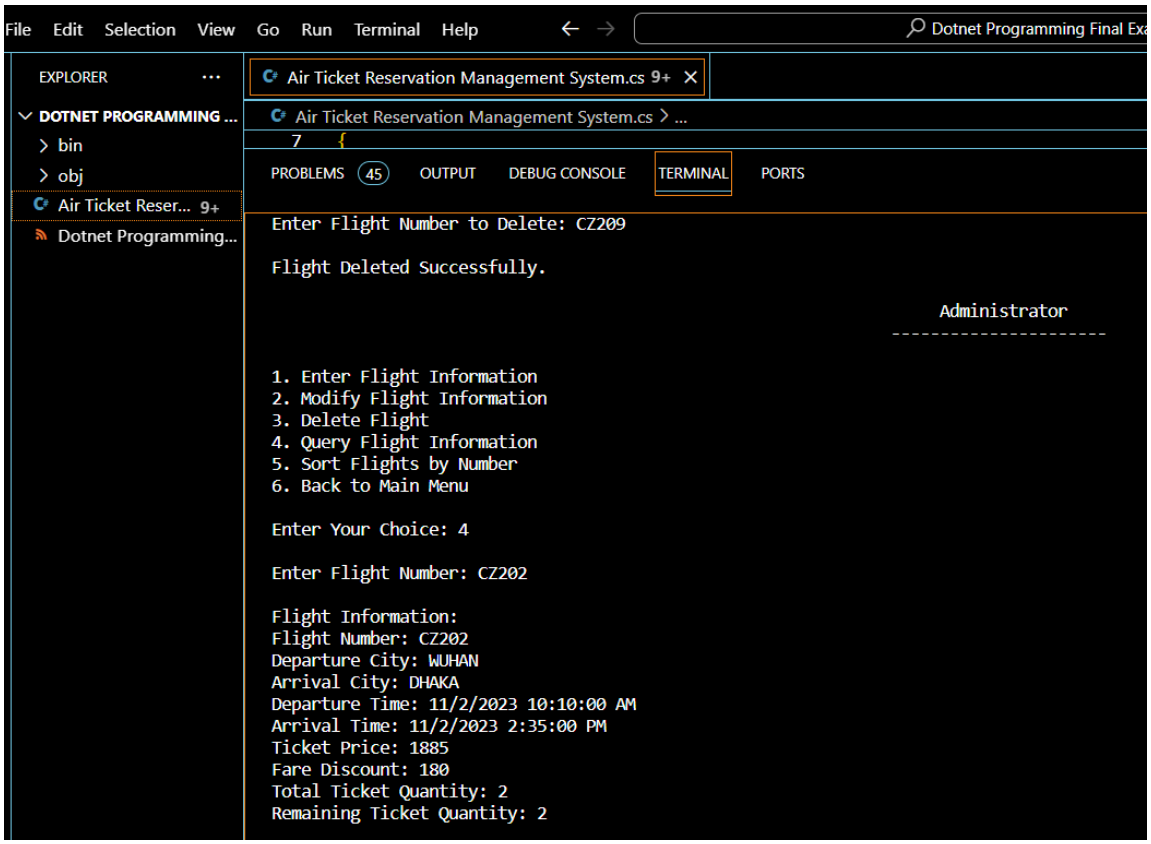

**Sorted Flights by Number**

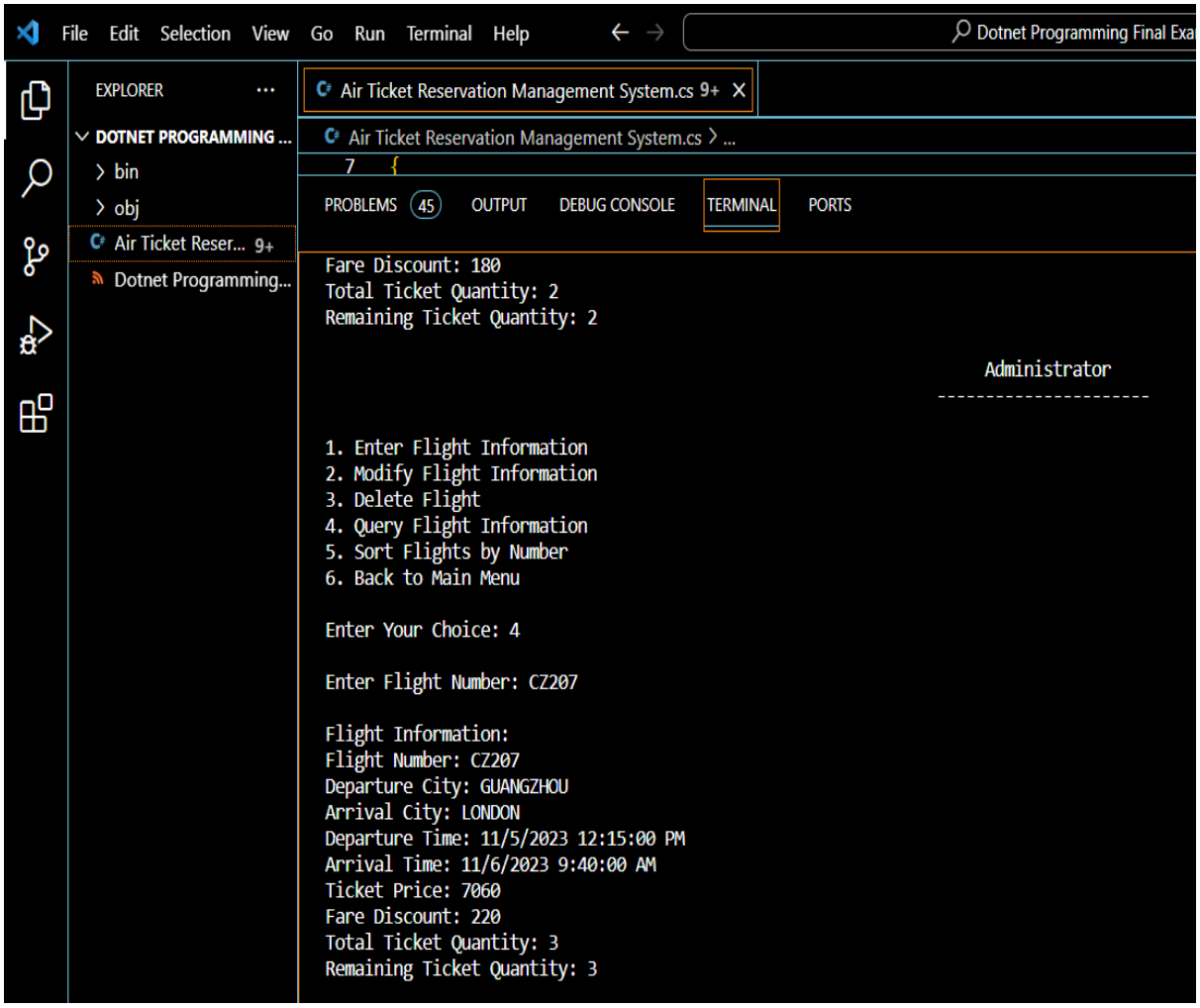
## 4.2.4 Delete Flight

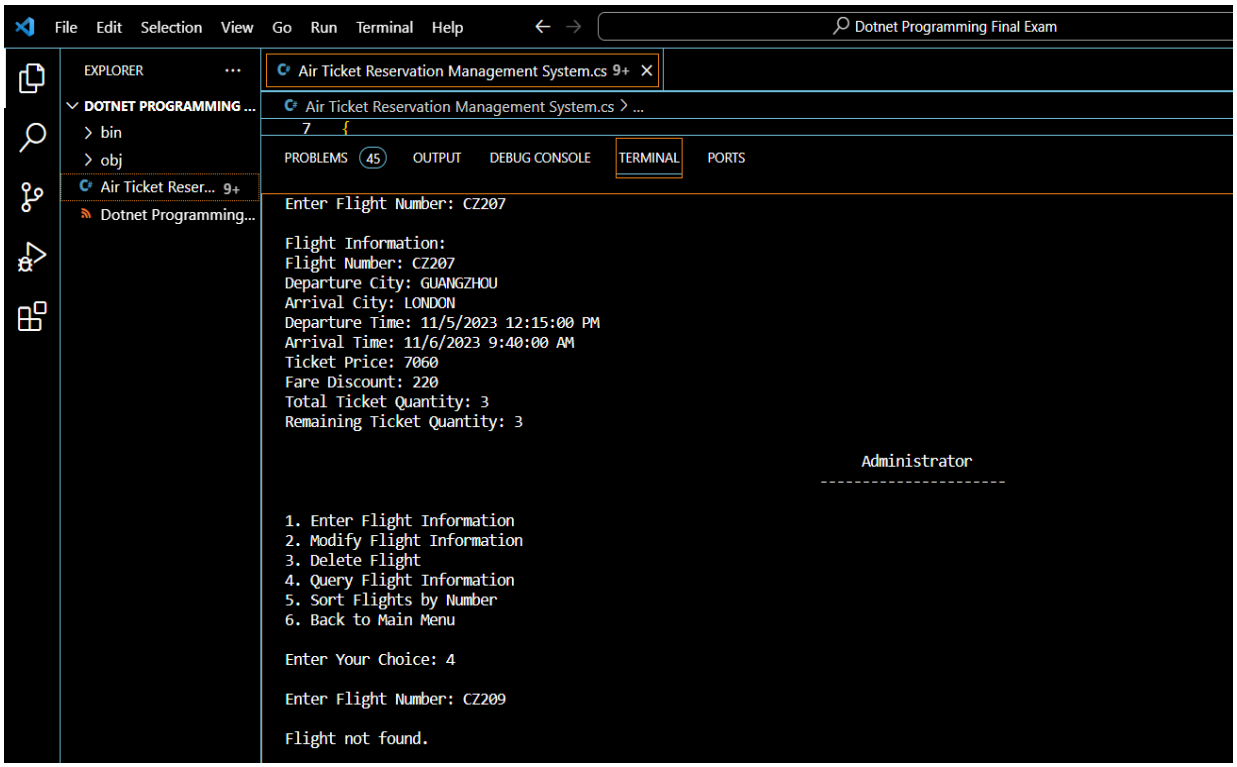


Otherwise,


## 4.2.5 Query Flight

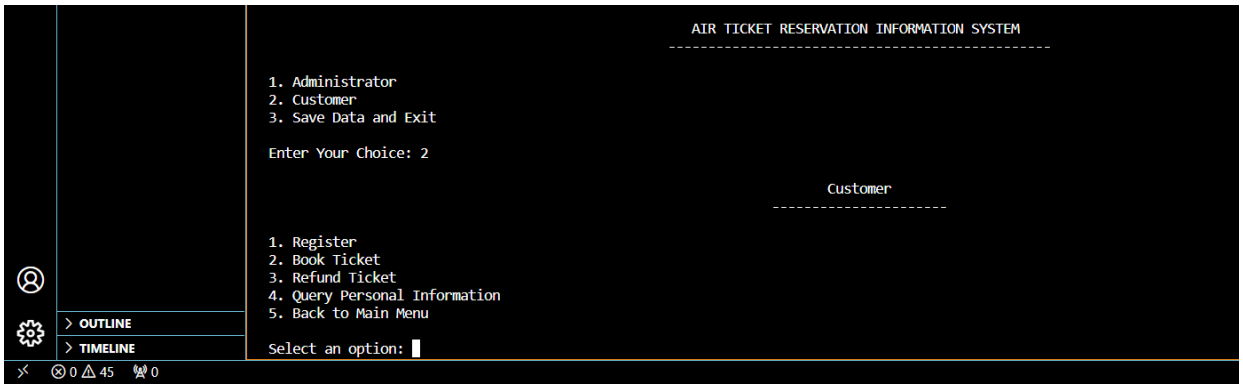



File Edit Selection View Go Run Terminal Help
Dotnet Programming Final Exa
EXPLORER
DOTNET PROGRAMMING ...
bin
obj
Air Ticket Reser... 9+
Dotnet Programming...
Air Ticket Reservation Management System.cs 9+
Air Ticket Reservation Management System.cs > ...
7 {
PROBLEMS 45 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Fare Discount: 180
Total Ticket Quantity: 2
Remaining Ticket Quantity: 2
Administrator
1. Enter Flight Information
2. Modify Flight Information
3. Delete Flight
4. Query Flight Information
5. Sort Flights by Number
6. Back to Main Menu
Enter Your Choice: 4
Enter Flight Number: CZ207
Flight Information:
Flight Number: CZ207
Departure City: GUANGZHOU
Arrival City: LONDON
Departure Time: 11/5/2023 12:15:00 PM
Arrival Time: 11/6/2023 9:40:00 AM
Ticket Price: 7060
Fare Discount: 220
Total Ticket Quantity: 3
Remaining Ticket Quantity: 3

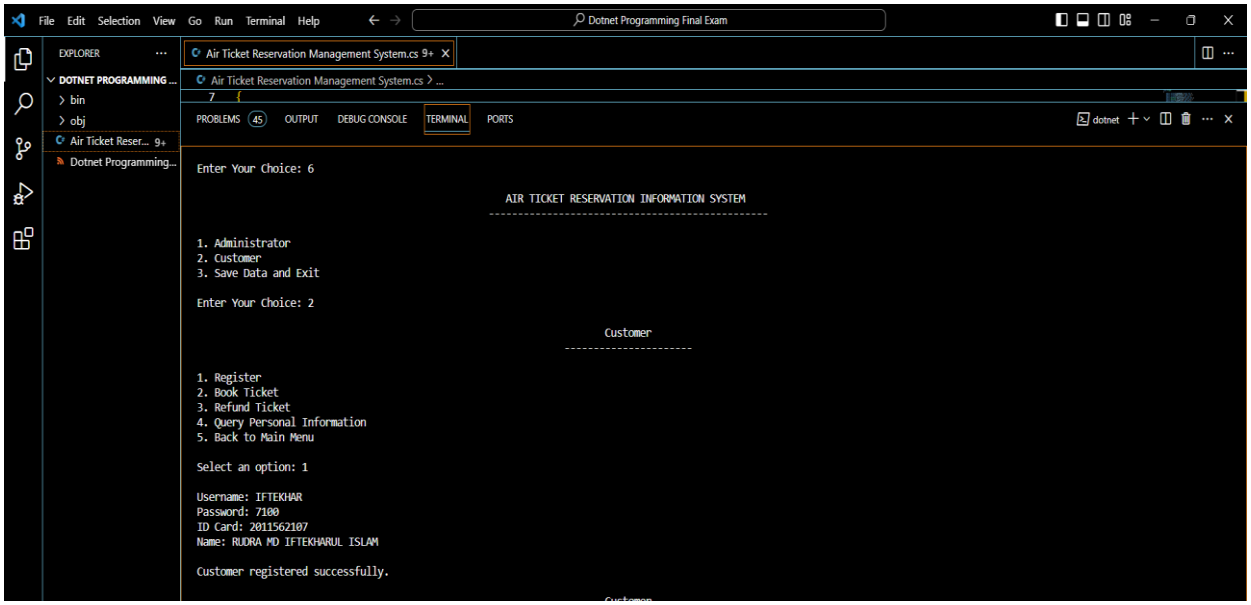



Otherwise,

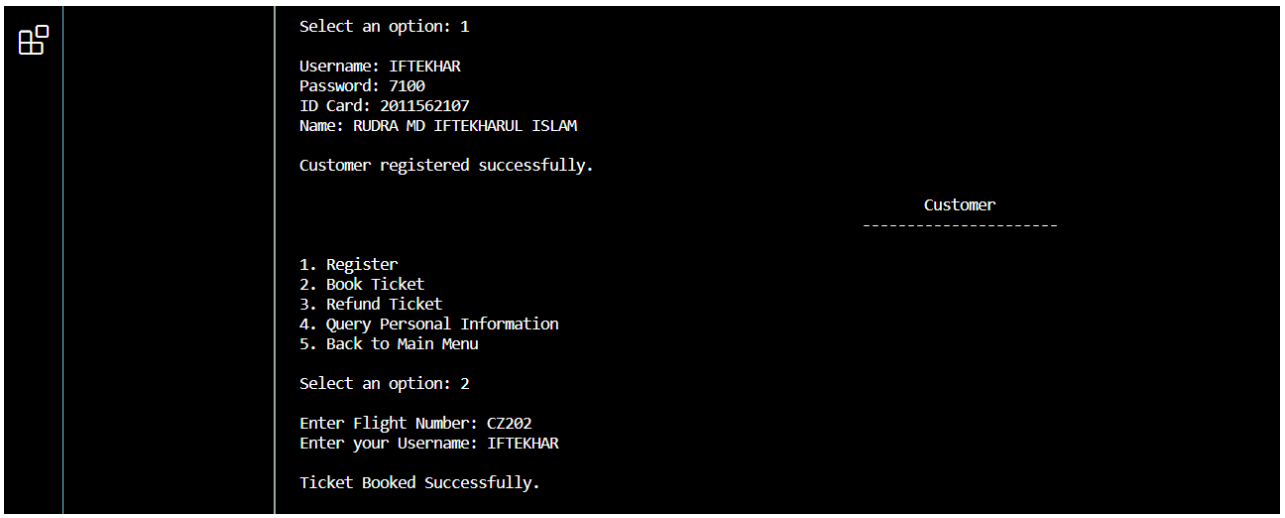

# 4.3 Customer Manu



### 4.3.1 Register Customer



## 4.3.2 Book Ticket

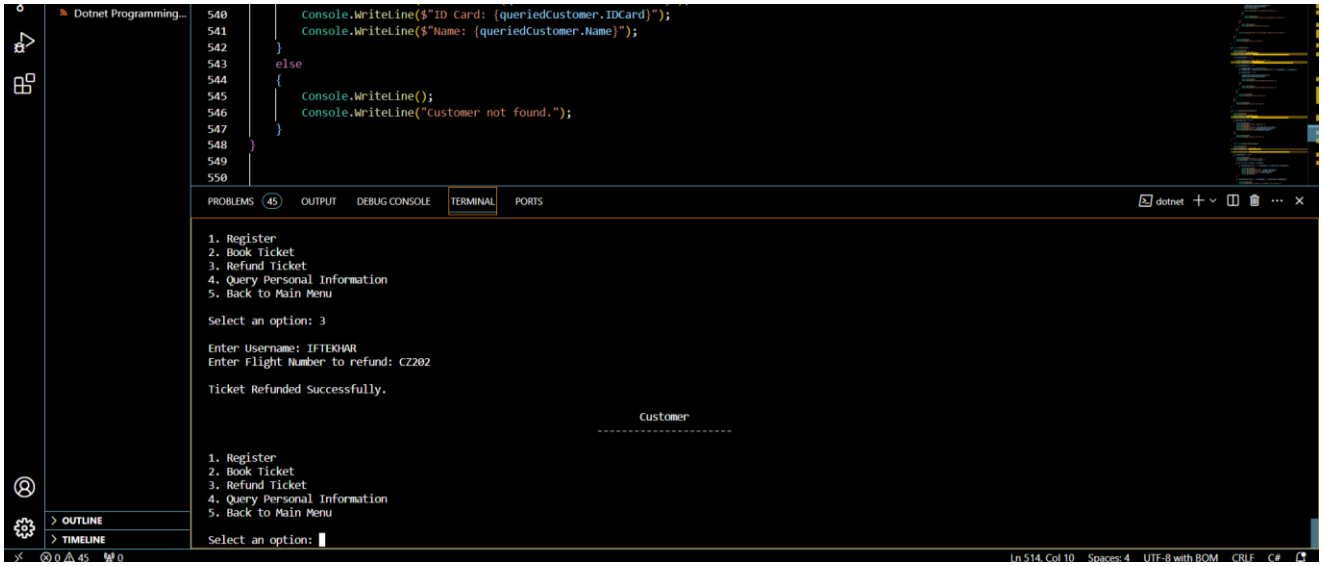



Otherwise,


## 4.3.3 Query Personal Information


Otherwise,


## 4.3.4 Refund Ticket



# Chapter 5: Conclusion

The Air Ticket Reservation Information System represents a comprehensive and efficient solution for managing flight information and customer interactions within the aviation industry. This project report has detailed the system's general purpose, significance, literature review, methodology, and its functional requirements. The implementation of these requirements, as demonstrated in the provided code, showcases a robust system with features designed to cater to both administrators and customers.

The project's general purpose is to offer a user-friendly platform for airlines and travelers to manage flight reservations effectively. The system simplifies flight information management, ticket booking, and customer interactions, streamlining the processes involved in air travel. The significance of this project lies in its potential to enhance the air travel experience. For airlines, it provides a comprehensive tool for managing flight details, including ticket pricing, bookings, and fare discounts. For customers, it simplifies the ticket booking process, allowing them to access flight information and make reservations with ease. The literature review conducted for this project emphasizes the existing demand for efficient air ticket reservation systems and the benefits they bring to the aviation industry. A review of similar systems highlights the need for security, accuracy, and user-friendliness in such applications. The methodology section describes the approach used in building the system, including the use of object-oriented programming, data storage and retrieval, and user interface design. The implementation of key functions for flight and customer management, booking, and refunds is detailed, demonstrating how the system fulfills the specified requirements. The system's requirements include basic information, such as its purpose and significance, as well as detailed descriptions of its implemented functions. These functions encompass flight management, customer management, menu systems for administrators and customers, data loading and saving, error handling, and security measures.

In conclusion, the Air Ticket Reservation Information System offers a practical and valuable solution for airlines and travelers in the aviation industry. Its robust features, coupled with an intuitive user interface, make it a powerful tool for both administrators and customers. The successful implementation of this project aligns with the growing demand for efficient and secure air ticket reservation systems and has the potential to greatly improve the air travel experience for all parties involved. As the aviation industry continues to evolve, systems like this one will play a pivotal role in enhancing the efficiency and convenience of air travel.