```
using System;
using System.Collections.Generic;

namespace Appointment_Booking_Application
{

    public class Program
    {
        public static void Main()
        {
            bool isValidAppointmentDate;
            AppointmentDateVerification a1 = new AppointmentDateVerification();

            try
            {
                PatientDetail patientDetail = new PatientDetail();


                string doctorname = string.Empty;

                Console.Write("Patient Name: ");
                patientDetail.PatientName = Console.ReadLine();
                Console.Write("Patient Age: ");
                patientDetail.PatientAge = Convert.ToInt32(Console.ReadLine());

                var departments = new PatientDetail().GetDepartments();
                Console.WriteLine("\nDepartments List\n");
                foreach (var item in departments)
                {
                    Console.WriteLine(item);
                }

                Console.Write("\nChoose the department number from the above list (1-5) : ");
                int option = Convert.ToInt32(Console.ReadLine());
                string department = departments[option - 1].Substring(2);
                bool isValidDoctor = true;
                do
                {


                    List<string> doctors = new PatientDetail().GetDoctors(option);

                    if (doctors.Count > 0)
                    {
                        Console.WriteLine("\nDoctors in the {0} department\n", department);
                        foreach (var item in doctors)
                        {
                            Console.WriteLine(item);
                        }

                        Console.Write("\nDoctor Name : ");

                        doctorname = Console.ReadLine();

                        if (!doctors.Contains(doctorname))
                        {
                            isValidDoctor = false;
                            Console.WriteLine("{0} not found in our list", doctorname);
                        }
                        else
                        {
                            isValidDoctor = true;
                        }

                    }

                } while (!isValidDoctor);

                do
                {
                    Console.Write("\nAppointment Request Date (MM/dd/yyyy) : ");
                    DateTime appointmentRequestDate = Convert.ToDateTime(Console.ReadLine());

                    //Validate the appointmentRequestDate and print appropriate message
                    string s = new
AppointmentDateVerification().CheckAppointmentRequestDate(appointmentRequestDate);
                    if (s == "Appointment Confirmed!")
                    {
                        Console.WriteLine(s);
                        Random rd = new Random();
                        int p_id = rd.Next(int.MaxValue, int.MinValue);
                        Console.WriteLine("Patient Id - " + p_id);
                        Console.WriteLine("Please Contact " + doctorname + " on " + appointmentRequestDate);
                        isValidAppointmentDate = true;
                    }
                    else
                    {
                        Console.WriteLine(s);
                        isValidAppointmentDate = false;
                    }


                } while (!isValidAppointmentDate);

            }
            catch (Exception e)
            {
                Console.WriteLine(e.Message);
            }
        }
    }
}



public class PatientDetail
{
    public string PatientName;
    public int PatientAge;
    public List<string> GetDepartments()
    {
        List<string> list = new List<string>();

        list.Add("ENT");
        list.Add("Gynecology");
        list.Add("Cardiology");
        list.Add("Neurology");
        list.Add("Nephrology");
        return list;
    }
    public List<string> GetDoctors(int option)
    {
        List<string> l1 = new List<string>();
        switch (option)
        {
            case 1:
                l1.Add("Dr. Murugadoss");
                l1.Add("Dr. Kalaivani");
                break;
            case 2:
                l1.Add("Dr. Abirami");
                l1.Add("Dr. Lakshmi");
                l1.Add("Dr. Revathi");
                break;
            case 3:
                l1.Add("Dr. Amudhan");
                l1.Add("Dr. Gunaseelan");
                l1.Add("Dr. Agarwal");
                break;
            case 4:
                l1.Add("Dr. Natarajan");
                l1.Add("Dr. Nanda");
                l1.Add("Dr. Keerthi");
                break;
            case 5:
                l1.Add("Dr. Ashirvatham");
                l1.Add("Dr. Cherian");
                l1.Add("Dr. Ram");
                break;
        }
        return l1;
    }
}

    public class AppointmentDateVerification
{
    public string CheckAppointmentRequestDate(DateTime appointmentRequestDate)
    {
        DateTime dt = DateTime.Today;
        DayOfWeek th = appointmentRequestDate.DayOfWeek;
        if (appointmentRequestDate > dt)
        {
            if (appointmentRequestDate.Year == dt.Year)
            {
                if (th != DayOfWeek.Monday)
                {
                    return "Appointment Confirmed!";
                }
                else
                {
                    return "Sorry!!! Appointment cannot be given on Monday!";
                }
            }
            else
            {
                return "Appointment Rejected, You can book appointment only for the current year!";
            }
        }
        else
        {
            return "Appointment Rejected, Date must be a future date!";
        }

    }
}
```

Appointment booking:

program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace AirlineTicketing  // Do not change the namespace name
{
    public class Program  // Do not change the class name
    {
        static void Main(string[] args) // Do not change the method name
        {
            //Implement the code here
            int nt;
            string td;
            int ns;
            string stbs;
            Console.WriteLine("Enter the number of tickets");
            nt = int.Parse(Console.ReadLine());
            List<Ticket>ticketList = new List<Ticket>();
            for(int i=1; i<=nt;i++)
            {
                td = Console.ReadLine();
                string[] details = td.Split(':');
                string name = details[0];
                string seatType = details[1];
                string seatId = details[2];

                if(TicketingDepartment.ValidateSeatId(seatId))
                {
                    Ticket ticket = new Ticket(name,seatType,seatId);
                    ticketList.Add(ticket);
                }
            }
            Console.WriteLine("Enter the number of seats to be searched");
            ns = int.Parse(Console.ReadLine());
            TicketingDepartment tdp;
            for( int i=1;i<= ns; i++)
            {
                Console.WriteLine("Enter seat type");
                stbs = Console.ReadLine();
                tdp = new TicketingDepartment(stbs,ticketList);
                Console.WriteLine(stbs+":"+tdp.SearchBySeatType(stbs));
            }

        }
    }
}
```

ticket.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace AirlineTicketing  // Do not change the namespace name
{
    public class Ticket  // Do not change the class name
    {
        //Implement the code here
        private string passangerName;
        private string seatType;
        private string seatId;

        public string PassangerName
        {
         get{ return passangerName; }
         set{ passangerName = value; }
        }

        public string SeatType
        {
            get{ return seatType;}
            set { seatType = value;}
        }

        public string SeatId
        {
            get{ return seatId;}
            set{ seatId = value;}
        }

        public Ticket(){}
        public Ticket( string passangerName,string seatType, string seatId)
        {
            this.passangerName = passangerName;
            this.seatType = seatType;
            this.seatId = seatId;
        }
        public string GetSeatType()
        {
            return seatType;
        }
        }
    }
```

TicketingDepartment.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace AirlineTicketing  // Do not change the namespace name
{
    public class TicketingDepartment  // Do not change the class name
    {
        //Implement the code here
        private string seatType;
        private List<Ticket>ticketList;

        public string SeatType
        {
            get{ return seatType;}
            set { seatType =value;}
        }

        public List<Ticket>TicketList
        {
```

```
    get{ return ticketList;}
    set{ ticketList = value;}
}

public TicketingDepartment(){}

public TicketingDepartment(string seatType, List<Ticket>ticketList)
{
    this.seatType= seatType;
    this.ticketList = ticketList;
}
public static bool ValidateSeatId(string seatId)
{
    try
    {
        if(seatId.Length !=3)
        {
            throw new SeatIdInvalidException("Invalid Seat Id");
        }
        char f_letter = seatId[0];
        if( f_letter !='A' && f_letter != 'B' && f_letter !='C' && f_letter !='D')
            throw new SeatIdInvalidException("Invalid Seat Id");
        char sL = seatId[1];
        char tL = seatId[2];
         if(!char.IsDigit(sL) || !char.IsDigit(tL))
            throw new SeatIdInvalidException("Invalid Seat Id");
        string number ="";
        if(sL =='0')
            number = seatId.Substring(2);
        else
            number = seatId.Substring(1);
        int num = int.Parse(number);
        if(!(num>=1 && num<=45))
            throw new SeatIdInvalidException("Invalid Seat Id");
        return true;
    }
    catch(Exception e)
    {
        return false;
    }
}
public int SearchBySeatType(string seatType)
{
    int count = 0;
    foreach(Ticket ticket in ticketList)
    {
```

```
                if( ticket.GetSeatType() == seatType)
                    count++;
            }
            return count;
        }
    }
}
```

SeatIdInvalidException.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace AirlineTicketing  // Do not change the namespace name
{
    public class SeatIdInvalidException:Exception  // Do not change the class name
    {
        //Implement the code here
        public SeatIdInvalidException(string message): base(message)
        {}
    }
}
```