```
public interface IAppointment
{
    public string PatientName { get; }
    public IEnumerable<DateTime> ProposedTimes { get; }
    public DateTime? SelectedAppointmentTime { set; }
}
public class MedicalScheduler
{
    public Dictionary<DateTime,string> Appointments { get; } =
         new Dictionary<DateTime, string>();
    public List<DateTime> FreeSlots { get; } = new List<DateTime>();
    public IEnumerable<Tuple<string,bool>>
     Schedule(IEnumerable<IAppointment> requests)
    {
        using(var it = requests.GetEnumerator())
        {
            bool check = false;
            while (it.MoveNext())
            {
                var patient = (IAppointment)it.Current;
                if(FreeSlots.Count == 0) yield return new Tuple<string, bool>
                     (patient.PatientName, false);
                else
                {
                    foreach (var slot in patient.ProposedTimes)
                    {
                        if (FreeSlots.Contains(slot))
                        {
                            FreeSlots.Remove((DateTime)slot);
                            Appointments.Add((DateTime)slot, patient.PatientName);
                            patient.SelectedAppointmentTime = (DateTime)slot;
                            yield return new Tuple<string, bool>
                                  (patient.PatientName, true);
                            check = true;
                            break;
                        }
                    }
                    if(!check)
                        yield return new Tuple<string, bool>(patient.PatientName,
                            false);
                }
            }
        }
    }
}

public class Appointment : IAppointment
{
    public string PatientName { get; set; }

    public IEnumerable<DateTime> ProposedTimes { get; set; }

    public DateTime? SelectedAppointmentTime { get; set; }

    public Appointment(string patientName, IEnumerable<DateTime> proposedTimes)
    {
        PatientName = patientName;
        ProposedTimes = proposedTimes;
    }
}
public class MedicalSchedulerTest
{
    [Test]
    public void NoFreeSlotFit()
```

```
{
    var ms = new MedicalScheduler();
    var giorgio = new Appointment("Giorgio", new DateTime[] {new (2022, 5, 3,
      11, 0, 0)});
    Assert.That(ms.Schedule(new []{ giorgio }),Is.EqualTo(
     new Tuple<string, bool>[] { new ("Giorgio", false) }));
}

[Test]
public void ScheduleOk()
{
    var ms = new MedicalScheduler();
    ms.FreeSlots.Add(new DateTime(2022, 8, 2, 8, 0, 0));
    ms.FreeSlots.Add(new DateTime(2022, 4, 16, 11, 0, 0));
    ms.FreeSlots.Add(new DateTime(2022, 4, 16, 10, 0, 0));
    ms.FreeSlots.Add(new DateTime(2022, 5, 3, 10, 0, 0));
    var adaAppointment = new DateTime[]
        {new DateTime(2022, 7, 2, 18, 0, 0),
            new DateTime(2022, 4, 16, 10, 0, 0)};
    var ada = new Appointment("Ada", adaAppointment);
    var schedule = ms.Schedule(new[] { ada }).ToArray();
    var ris = ms.Appointments[new DateTime(2022, 4, 16, 10, 0, 0)];
    Assert.That(ris,Is.EqualTo("Ada"));
}

[Test]
public void SheduleWithAppointments()
{
    var ugoAppointment = new DateTime[]
    {new DateTime(2022, 3, 12, 15, 0, 0),
        new DateTime(2022, 3, 13, 15, 0, 0)};
    var ugo = new Appointment("Ugo", ugoAppointment);

    var ms = new MedicalScheduler();

    ms.FreeSlots.Add(new DateTime(2022, 8, 20, 18, 0, 0));
    ms.FreeSlots.Add(new DateTime(2022, 6, 6, 16, 0, 0));
    ms.FreeSlots.Add(new DateTime(2022, 3, 13, 15, 0, 0));
    ms.FreeSlots.Add(new DateTime(2022, 3, 3, 15, 0, 0));

    ms.Appointments.Add(new DateTime(2022, 8, 13, 10, 0, 0), "Franco");
    ms.Appointments.Add(new DateTime(2022, 6, 6, 15, 0, 0), "Luca");
    ms.Appointments.Add(new DateTime(2022, 6, 6, 17, 0, 0), "Paola");

    var schedule = ms.Schedule(new[] { ugo }).ToArray();

    Dictionary<DateTime,string> risAppointments =
           new Dictionary<DateTime,string>();
    risAppointments.Add(new DateTime(2022, 8, 13, 10, 0, 0), "Franco");
    risAppointments.Add(new DateTime(2022, 6, 6, 15, 0, 0), "Luca");
    risAppointments.Add(new DateTime(2022, 6, 6, 17, 0, 0), "Paola");
    risAppointments.Add(new DateTime(2022, 3, 13, 15, 0, 0), "Ugo");

    Assert.Multiple(() =>
    {
        Assert.That(ms.FreeSlots,
            Is.EqualTo(new[]
            {
                new DateTime(2022, 8, 20, 18, 0, 0) ,
                new DateTime(2022, 6, 6, 16, 0, 0) ,
                new DateTime(2022, 3, 3, 15, 0, 0)
            }));
        Assert.That(ms.Appointments,Is.EqualTo(risAppointments));
    });
    }


    [Test]
    public void NoFreeSlotFitMock()
    {
        var ms = new MedicalScheduler();

        var mc = new Mock<IAppointment>();

        mc.Setup(x => x.PatientName).Returns("Giorgio");
        mc.Setup(x => x.ProposedTimes).Returns(
         new DateTime[] { new DateTime(2022, 5, 3, 11, 0, 0) });

        Assert.That(ms.Schedule(new[] { mc.Object }),
                Is.EqualTo(new Tuple<string, bool>[] {
               new Tuple<string, bool>("Giorgio", false) }));
    }


}
```