

Assignment 3: GetWellApp - Healthcare Management System

Objective:

Develop a simple healthcare management system using C# concepts like Aggregation and Association.

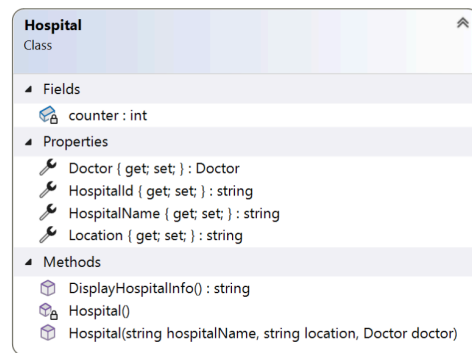
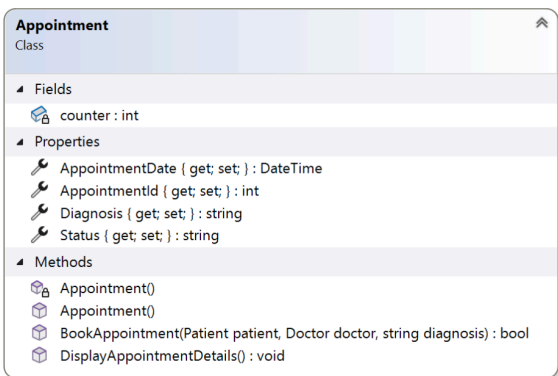
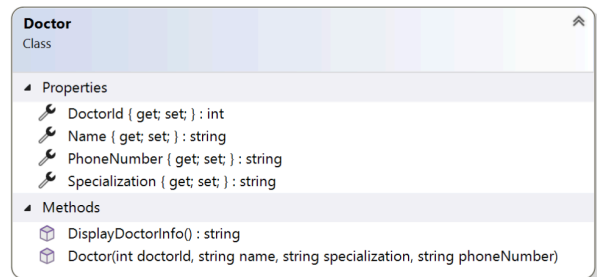
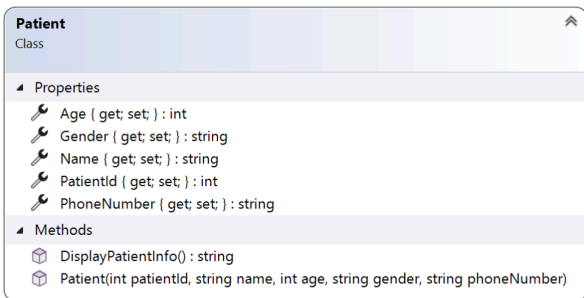
Solution Structure:

Solution Name: GetWellApp

Class Library: GetWellAppBusinessLayer

Console Application: GetWellAppConsoleApp

Class Diagram:



Assignment Steps:

Step 1: Create the Visual Studio Solution

1. Open **Visual Studio** and create a new **solution** named `GetWellApp`.
 2. Add a **Class Library project** named `GetWellAppBusinessLayer`.
 3. Add a **Console Application project** named `GetWellConsoleApp`.
 4. Add a reference to `GetWellAppBusinessLayer` in `GetWellConsoleApp`.
-

Step 2: Create Classes in `GetWellAppBusinessLayer`

1. Create **Patient** Class

- **Purpose:** Represents a patient in the healthcare system.
 - **Properties:**
 - `PatientId` (int)
 - `Name` (string)
 - `Age` (int)
 - `Gender` (string)
 - `PhoneNumber` (string)
 - **Constructor:** Initialize `PatientId`, `Name`, `Age`, `Gender`, `PhoneNumber`.
 - **Method:** `string DisplayPatientInfo()` → Returns patient details as a concatenated string.
-

2. Create **Doctor** Class

- **Purpose:** Represents a doctor in the hospital.
- **Properties:**

- **DoctorId** (int)
 - **Name** (string)
 - **Specialization** (string)
 - **PhoneNumber** (string)
 - **Constructor:** Initialize **DoctorId**, **Name**, **Specialization**, **PhoneNumber**.
 - **Method:** **string DisplayDoctorInfo()** → Returns doctor details as a concatenated string.
-

3. Create **Hospital** Class (Aggregation with Doctor)

- **Purpose:** Represents a hospital that has a doctor.
 - **Properties:**
 - **HospitalId** (string) - auto-generated
 - **Name** (string)
 - **Location** (string)
 - **Doctor** (Doctor)
 - **Field:**
 - counter - static variable to auto-generate HospitalId like H1001, H1002, H1003 etc.
 - **Static constructor:** Initialize counter variable appropriately.
 - **Constructor:** Initialize **HospitalId**, **Name**, **Location**, and assign a **Doctor**.
 - **Method:** **string DisplayHospitalInfo()** → Returns hospital details as a concatenated string.
-

4. Create **Appointment** Class (Association between Appointment, Doctor and Patient)

- **Purpose:** Represents an appointment between a **patient** and a **doctor**.
- **Properties:**

- `AppointmentId` (int)
 - `AppointmentDate` (DateTime)
 - `Diagnosis` (string)
 - `Status` (string)
 - **Field:**
 - counter - static variable to auto-generate AppointmentID like 501, 502, 503 etc.
 - **Static Constructor:** Initialize counter variable appropriately
 - **Parameterless Constructor** - No logic needed
 - **Methods:**
 - `void DisplayAppointmentDetails()` → Prints appointment details using Console.WriteLine statements.
 - `bool BookAppointment(Patient patient, Doctor doctor, string diagnosis)`
 - Books a new appointment for the given patient with the doctor.
 - Check if the patient and doctor are valid (not null)
 - If yes
 - Auto-generate the AppointmentID using the counter and assign it.
 - Assign the diagnosis value to the Diagnosis property.
 - Set the AppointmentDate for 2 hours after the current date and time.
 - Set the Status as "Scheduled".
 - Display the message *"Appointment for <patient name> booked successfully with Dr. <doctor name>."*
 - Invoke the `DisplayAppointmentDetails()` method to print the appointment details.
 - Return true from the method.
 - Else, return false from the method.
-

Step 3: Implement Business Logic in GetWellAppConsoleApp

1. Instantiate Objects:

- Create a **Patient**.
- Create a **Doctor**.
- Create a **Hospital** and associate it with the Doctor.

2. Call Methods:

- Display **patient details** using `DisplayPatientInfo()`.
 - Display **doctor details** using `DisplayDoctorInfo()`.
 - Display **hospital details** using `DisplayHospitalInfo()`.
 - Book an appointment and display its details for the patient with the doctor using the `BookAppointment()`.
 - If the `BookAppointment()` returns true, display "Success", else display "Failed".
-

Step 4: Program.cs code

```
static void Main(string[] args)
{
    Patient patient = new Patient(201, "John Doe", 45, "Male",
    "9998887776");

    Doctor doctor = new Doctor(101, "Smith Jones", "Cardiologist",
    "9876543210");

    Hospital hospital = new Hospital("City Hospital", "Church Street",
    doctor);

    Console.WriteLine(patient.DisplayPatientInfo());
    Console.WriteLine(doctor.DisplayDoctorInfo());
    Console.WriteLine(hospital.DisplayHospitalInfo());

    Appointment appointment = new Appointment();
```

```
bool status = appointment.BookAppointment(patient, doctor,  
"Diabetes");
```

```
if (status)  
{  
    Console.WriteLine("Success");  
}  
else  
{  
    Console.WriteLine("Failed");  
}  
}
```

Step 5: Sample Console Output (Expected Behavior)

```
Patient Details: 201 John Doe 45 Male 9998887776  
Doctor Details: 101 Smith Jones Cardiologist 9876543210  
Hospital Details: H1001 City Hospital Church Street Smith Jones  
Appointment for John Doe booked successfully with Dr. Smith Jones.  
Appointment Details:  
Appointment ID: 501  
Appointment Date: 30-01-2025 19:15:51  
Diagnosis: Diabetes  
Success
```