

وثيقة توصيف متطلبات لنظام حجز وإدارة مواعيد في مركز طبي

اعداد : زينب اسمندر

اشراف : أ . تمام

1.1 المقدمة :

في العصر الحديث، أصبح الاعتماد على الأنظمة الإلكترونية لإدارة المواعيد الطبية أمرًا ضروريًا لتحسين جودة الخدمة وتقليل الازدحام في المراكز الطبية. يهدف هذا المشروع إلى تطوير نظام متكامل لإدارة مواعيد المرضى في العيادات والمراكز الطبية، بحيث يسهل على المرضى حجز المواعيد، وعلى الإدارة تنظيم عمل الأطباء والموظفين.

يوفر النظام بيئة آمنة وسهلة الاستخدام لإدارة المرضى، الأطباء، المواعيد، والدفع، مع مراعاة معايير الأمان والأداء وقابلية التوسع.

1.2 أهداف المشروع :

1. توفير نظام متكامل لإدارة المواعيد الطبية:

تنظيم حجوزات المرضى وجدولة الأطباء بشكل فعال

2. تحسين تجربة المستخدم:

واجهة سهلة للمرضى لحجز المواعيد ومتابعتها

3. ضمان أمان البيانات الطبية:

حماية معلومات المرضى باستخدام تقنيات التشفير

4. تقليل الازدحام وسوء التنظيم:

تنظيم أوقات العمل وتقليل الانتظار

5. دعم التوسع المستقبلي:

إمكانية إضافة عيادات وأطباء جدد بسهولة

6. الاعتماد على أنماط التصميم البرمجي (Design Patterns) :

لضمان بنية قوية وقابلة للصيانة

1.3 المتطلبات الوظيفية :

1. إدارة المرضى:

- تسجيل المرضى وإنشاء حسابات

- تحديث المعلومات الشخصية والطبية

- عرض سجل المواعيد السابقة

2. إدارة الأطباء:

- إضافة وتعديل بيانات الأطباء

- تحديد اختصاص الطبيب وساعات الدوام

- تتبع توفر الطبيب

3. إدارة المواعيد:

- البحث عن المواعيد المتاحة

- حجز موعد جديد

- تعديل أو إلغاء الموعد

- التحقق من تعارض المواعيد

4. الدفع والفوترة:

- دعم الدفع الإلكتروني

- إصدار فواتير للمرضى

- احتساب تكلفة الكشف والخدمات

5. إدارة المستخدمين (الموظفين):

- إنشاء حسابات للموظفين

- تحديد الصلاحيات (إدارة – استقبال – طبيب)

- مراقبة العمليات

6. الإشعارات والتنبيهات:

- إرسال تأكيد الموعد عبر البريد أو الرسائل

- تذكير المريض قبل الموعد

- إشعارات بتغييرات المواعيد

7. التقارير والتحليلات:

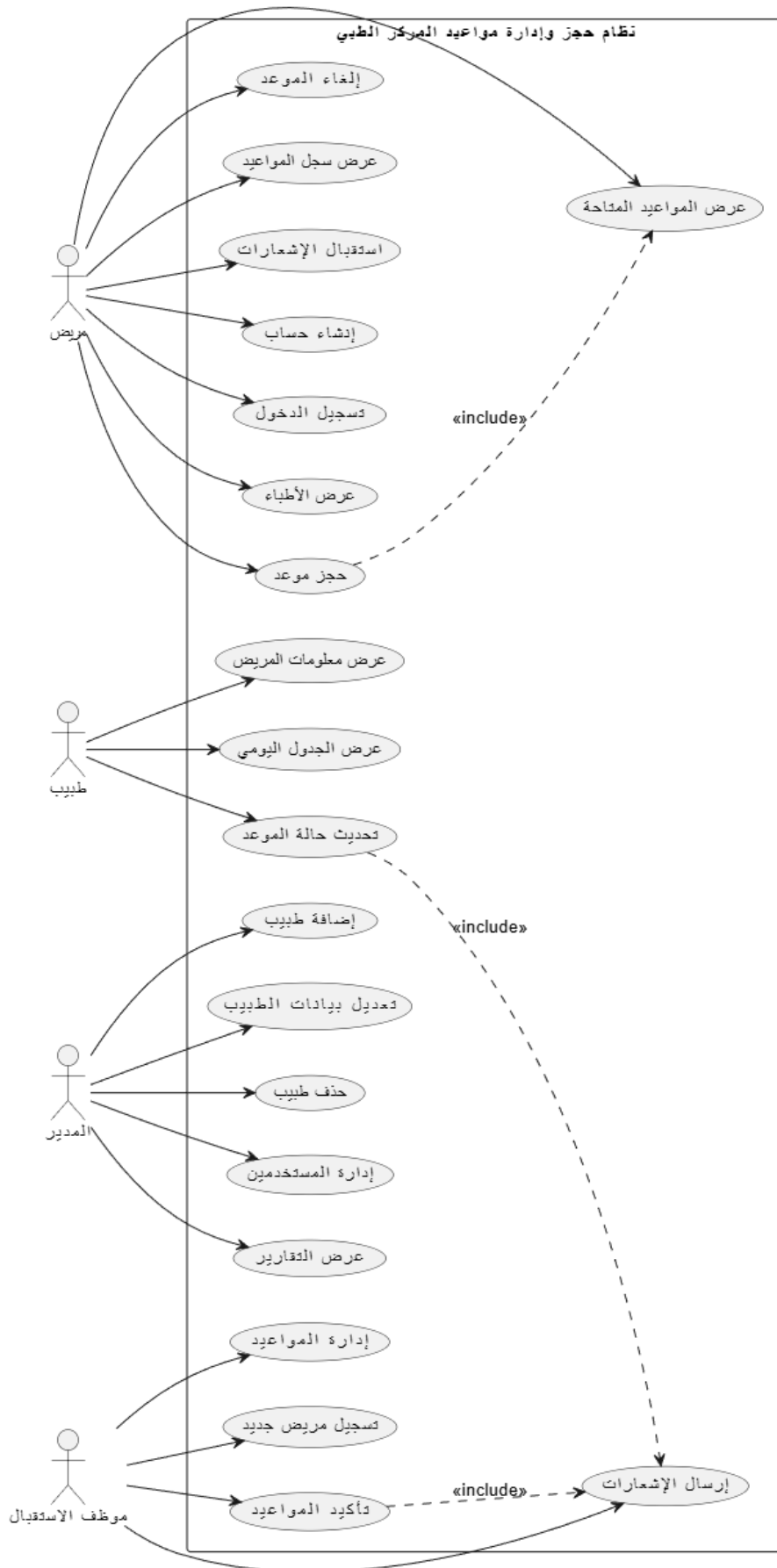
- تقارير عن عدد المواعيد

- تقارير عن دخل المركز
- إحصائيات عن أكثر الاختصاصات طلبًا

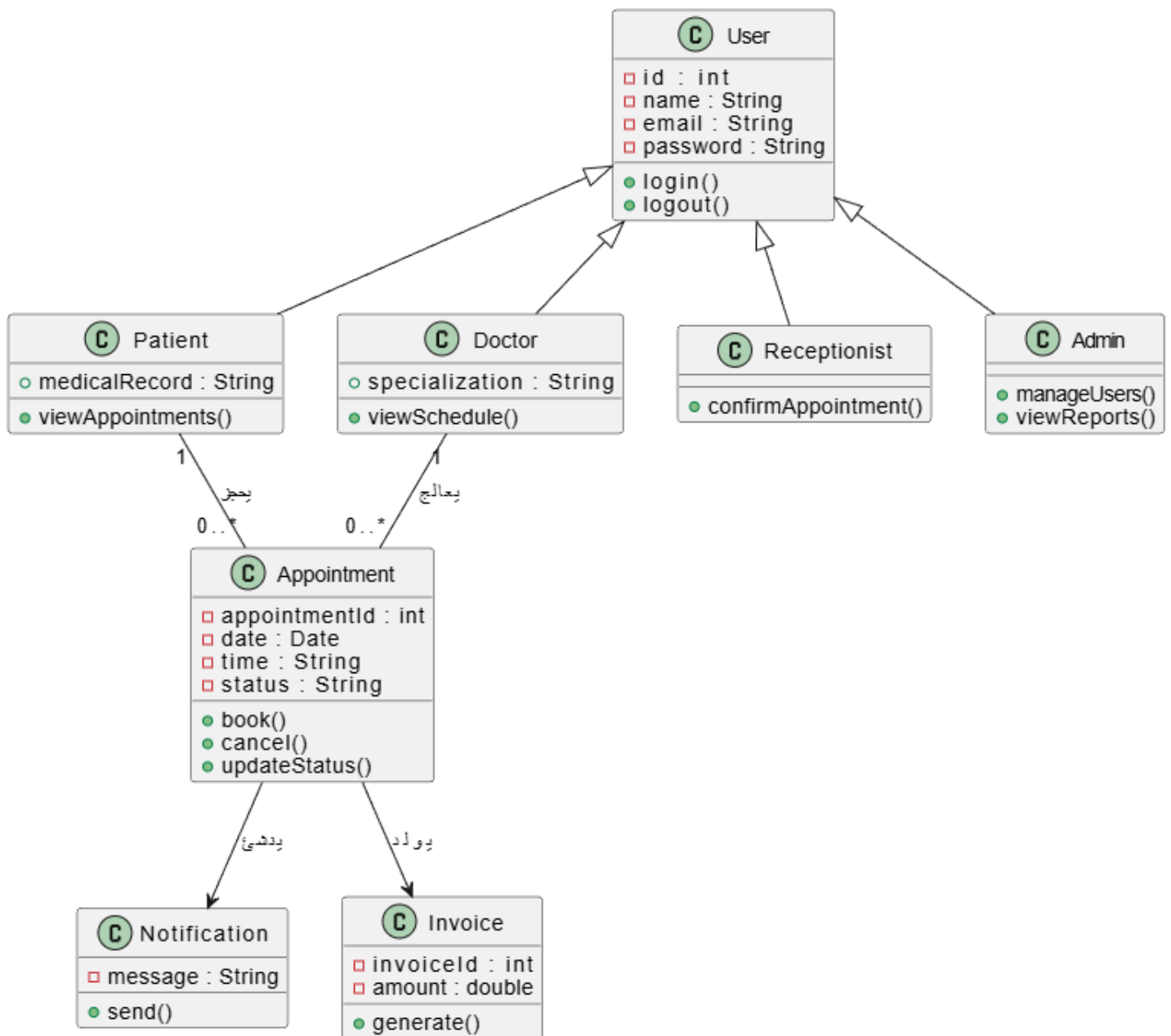
1.4 المتطلبات غير الوظيفية:

- الأداء: يجب أن يكون النظام سريع الاستجابة
- الأمان: تشفير بيانات المرضى
- القابلية للتوسع: دعم عدد كبير من المرضى
- سهولة الاستخدام: واجهة واضحة
- التوافق: يعمل على المتصفح والهاتف
- الصيانة: سهولة التحديث والتطوير

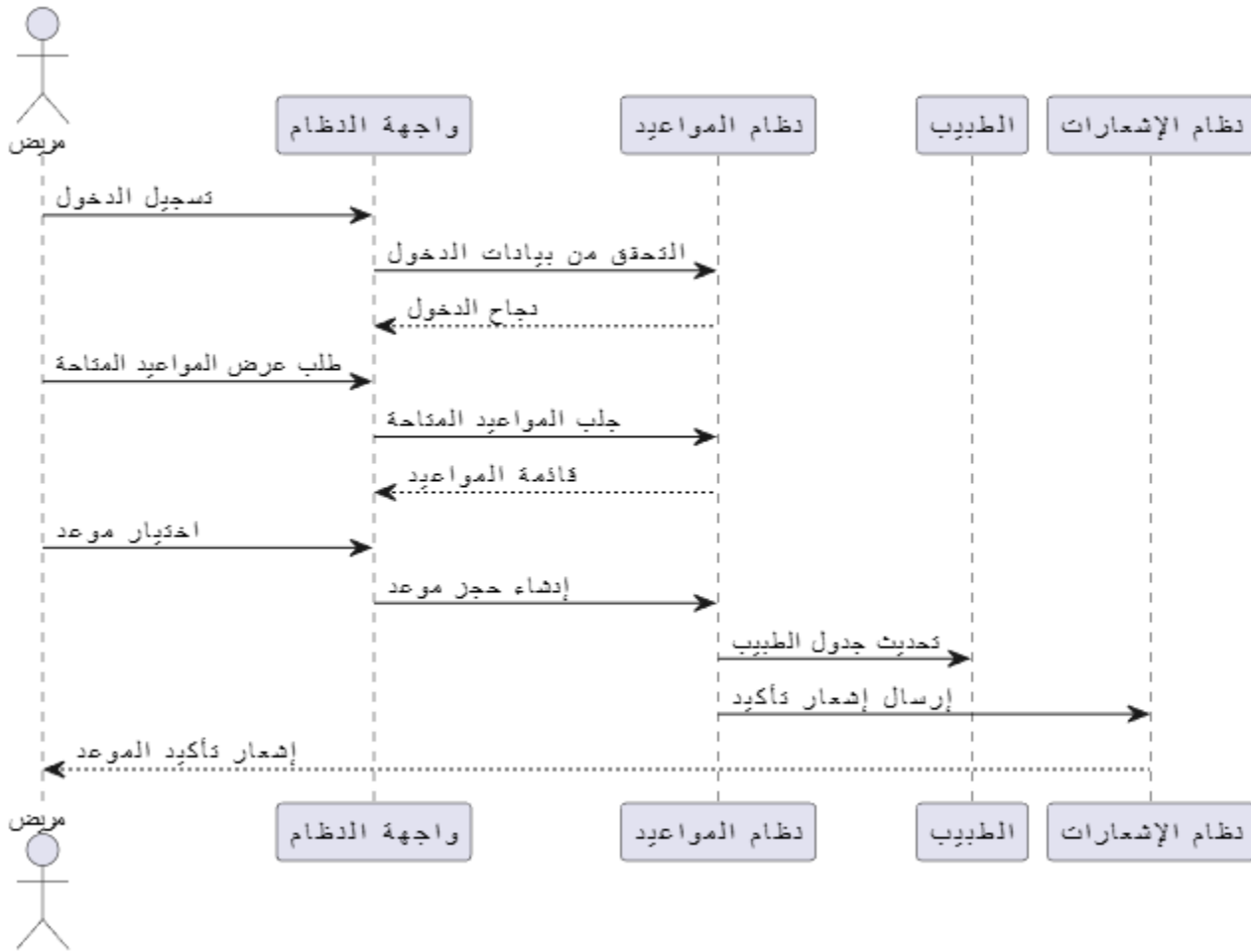
2.1 usecase diagram :



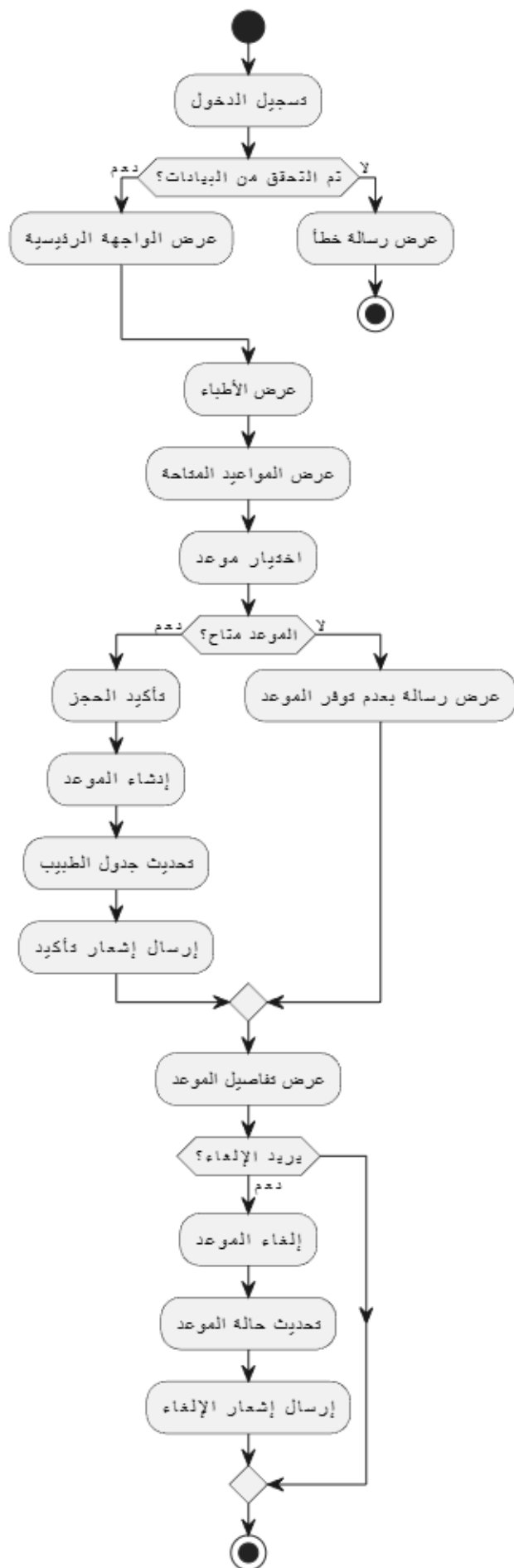
2.2 Class Diagram:



3.2 Sequence Diagram :



4.2Activity Diagram:



4. Design Patterns:

observer Pattern.1

لماذا اخترناه؟

لإرسال إشعارات تلقائية عند تغيير حالة الموعد.

تطبيقه في مشروعنا:

عند تأكيد أو تعديل أو إلغاء الموعد، يتم إشعار:

- المريض

- الطبيب

- موظف الاستقبال

بدون الحاجة للاستعلام الدوري.

Factory Pattern.2

لماذا اخترناه؟

لإنشاء أنواع مختلفة من المستخدمين بطريقة مرنة.

تطبيقه في مشروعنا:

لدينا أنواع:

- Patient

- Doctor

- Receptionist

- Admin

بدلاً من:

`new Doctor()`

نستخدم:

`UserFactory.createUser("Doctor")`

Decorator Pattern.3

لماذا اخترناه؟

لإضافة خدمات إضافية على الموعد بدون تعديل الكود الأساسي.

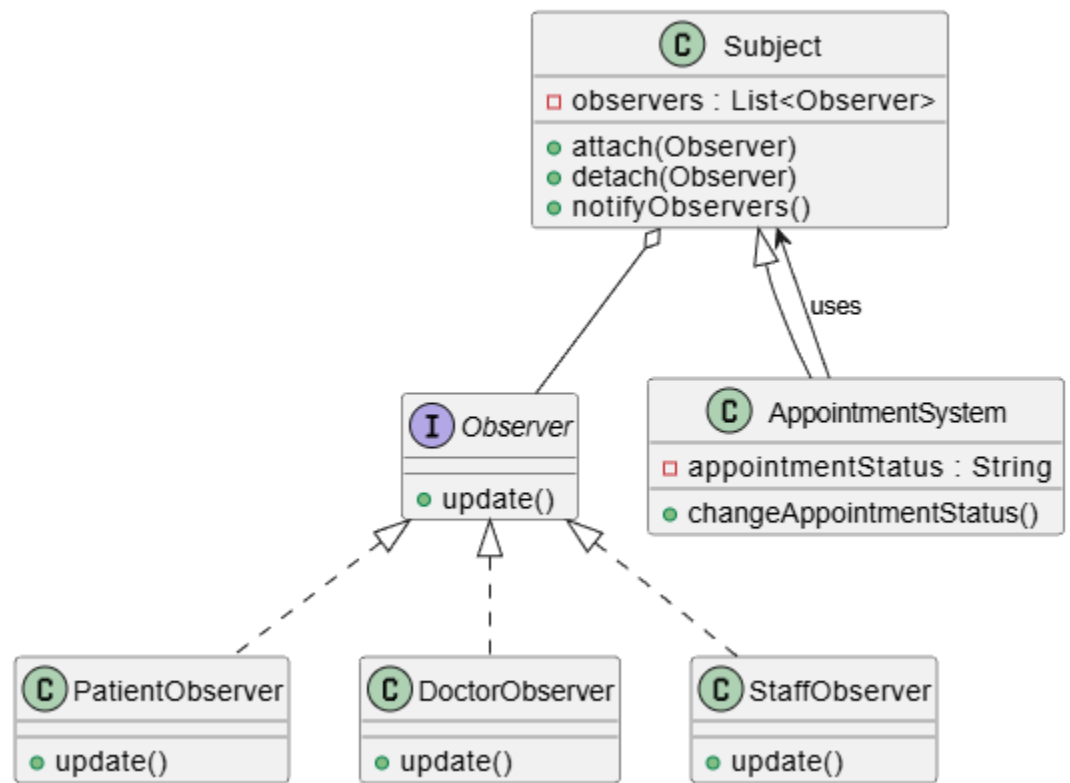
تطبيقه في مشروعنا:

يمكن إضافة خدمات مثل:

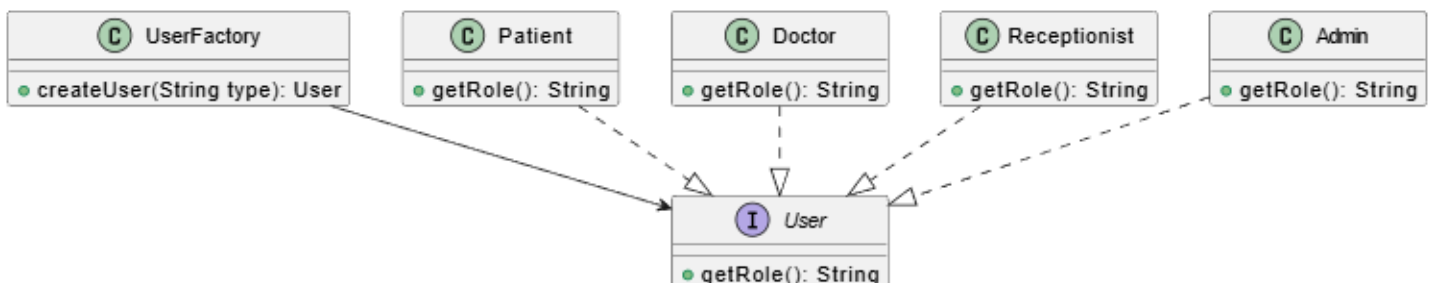
- فحص مخبري
- تصوير شعاعي
- استشارة إضافية

بدلاً من إنشاء فئات كثيرة، نستخدم Decorator لإضافة الخدمات ديناميكياً

Observer Pattern .1



Factory Pattern.2



Decorator Pattern.3

