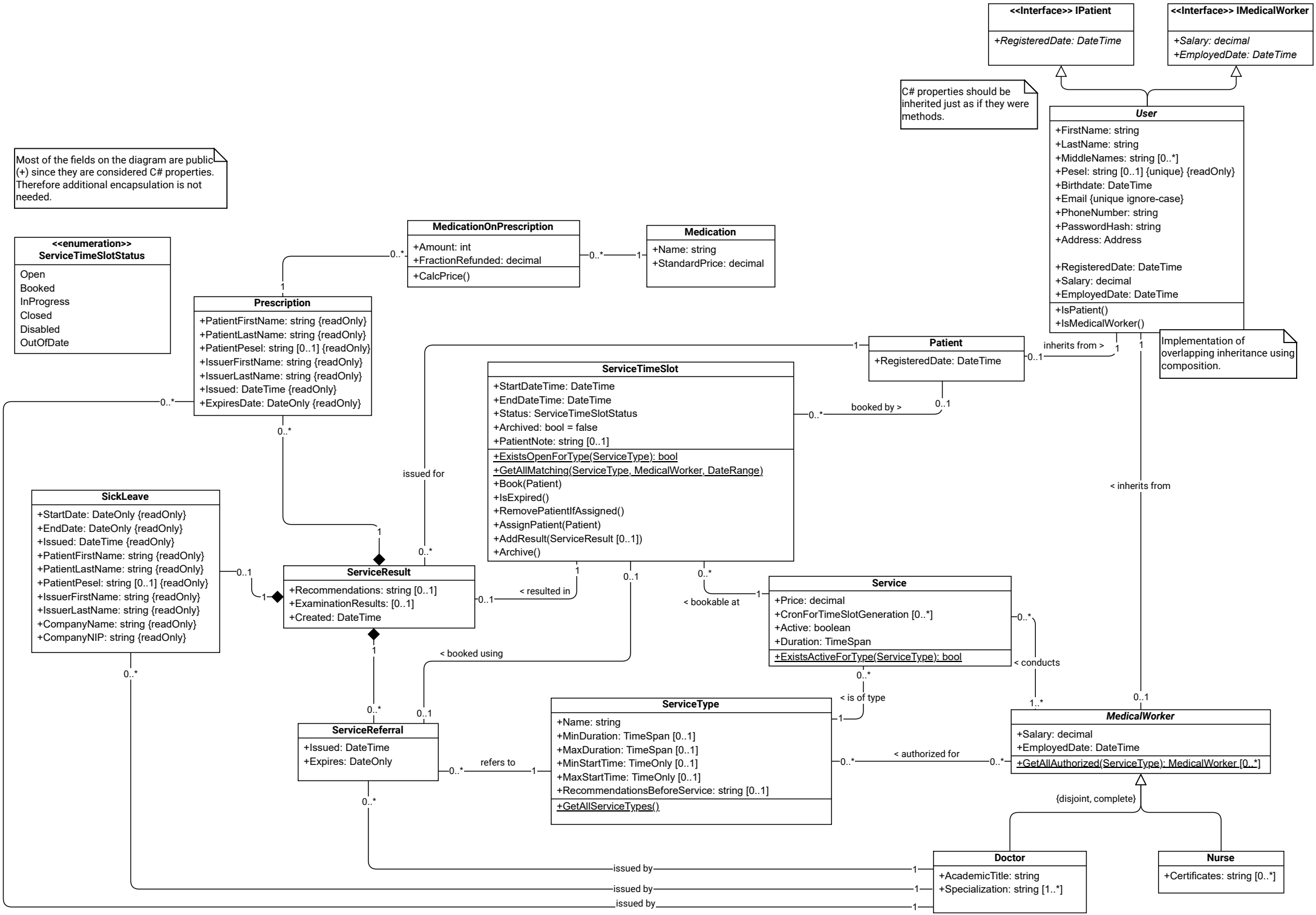
<<Interface>> IPatient
+RegisteredDate: DateTime
<<Interface>> IMedicalWorker
+Salary: decimal
+EmployedDate: DateTime
C# properties should be inherited just as if they were methods.
User
+FirstName: string
+LastName: string
+MiddleNames: string [0..*]
+Pesel: string [0..1] {unique} {readOnly}
+Birthdate: DateTime
+Email {unique ignore-case}
+PhoneNumber: string
+PasswordHash: string
+Address: Address
+RegisteredDate: DateTime
+Salary: decimal
+EmployedDate: DateTime
+IsPatient()
+IsMedicalWorker()
Implementation of overlapping inheritance using composition.
Most of the fields on the diagram are public (+) since they are considered C# properties. Therefore additional encapsulation is not needed.
<<enumeration>>
ServiceTimeSlotStatus
Open
Booked
InProgress
Closed
Disabled
OutOfDate
MedicationOnPrescription
+Amount: int
+FractionRefunded: decimal
+CalcPrice()
Medication
+Name: string
+StandardPrice: decimal
Prescription
+PatientFirstName: string {readOnly}
+PatientLastName: string {readOnly}
+PatientPesel: string [0..1] {readOnly}
+IssuerFirstName: string {readOnly}
+IssuerLastName: string {readOnly}
+Issued: DateTime {readOnly}
+ExpiresDate: DateOnly {readOnly}
Patient
+RegisteredDate: DateTime
inherits from >
< inherits from
ServiceTimeSlot
+StartDateTime: DateTime
+EndDateTime: DateTime
+Status: ServiceTimeSlotStatus
+Archived: bool = false
+PatientNote: string [0..1]
+ExistsOpenForType(ServiceType): bool
+GetAllMatching(ServiceType, MedicalWorker, DateRange)
+Book(Patient)
+IsExpired()
+RemovePatientIfAssigned()
+AssignPatient(Patient)
+AddResult(ServiceResult [0..1])
+Archive()
booked by >
issued for
SickLeave
+StartDate: DateOnly {readOnly}
+EndDate: DateOnly {readOnly}
+Issued: DateTime {readOnly}
+PatientFirstName: string {readOnly}
+PatientLastName: string {readOnly}
+PatientPesel: string [0..1] {readOnly}
+IssuerFirstName: string {readOnly}
+IssuerLastName: string {readOnly}
+CompanyName: string {readOnly}
+CompanyNIP: string {readOnly}
ServiceResult
+Recommendations: string [0..1]
+ExaminationResults: [0..1]
+Created: DateTime
< resulted in
< bookable at
Service
+Price: decimal
+CronForTimeSlotGeneration [0..*]
+Active: boolean
+Duration: TimeSpan
+ExistsActiveForType(ServiceType): bool
< conducts
< is of type
< booked using
ServiceReferral
+Issued: DateTime
+Expires: DateOnly
refers to
ServiceType
+Name: string
+MinDuration: TimeSpan [0..1]
+MaxDuration: TimeSpan [0..1]
+MinStartTime: TimeOnly [0..1]
+MaxStartTime: TimeOnly [0..1]
+RecommendationsBeforeService: string [0..1]
+GetAllServiceTypes()
< authorized for
MedicalWorker
+Salary: decimal
+EmployedDate: DateTime
+GetAllAuthorized(ServiceType): MedicalWorker [0..*]
{disjoint, complete}
issued by
Doctor
+AcademicTitle: string
+Specialization: string [1..*]
Nurse
+Certificates: string [0..*]