

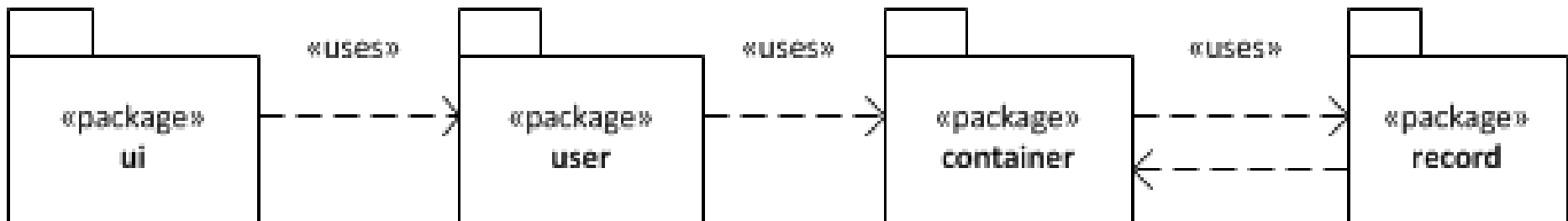
Software Engineering and Design

Task 7

Domain Model & Sequence Diagram

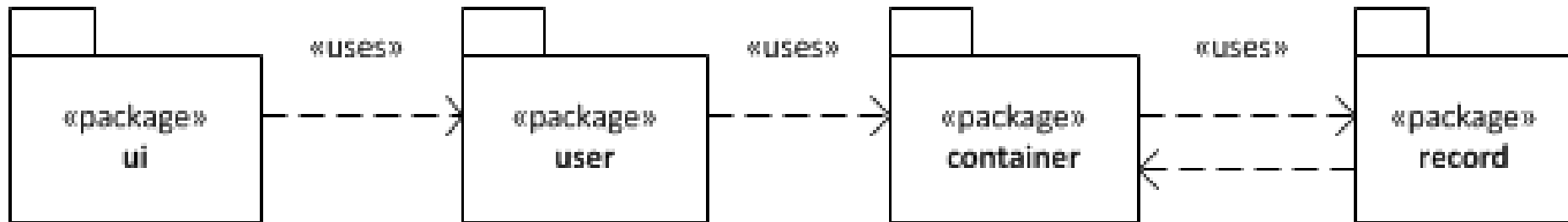
Packagestruktur

- * Ui-Package: Alle Klassen des User-Interfaces
- * User-Package: Klasse zur Verwaltung eines eingeloggten Benutzers (Psychiater, Psychologe, ...)



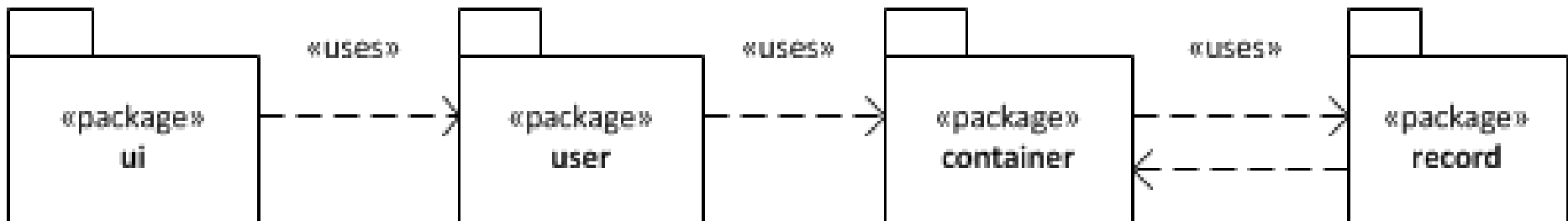
Packagestruktur

- * DataRecord-Package: Kapselt den Datensatz eines bestimmten Typs (z. B. Patienten, Diagnosen, ...)

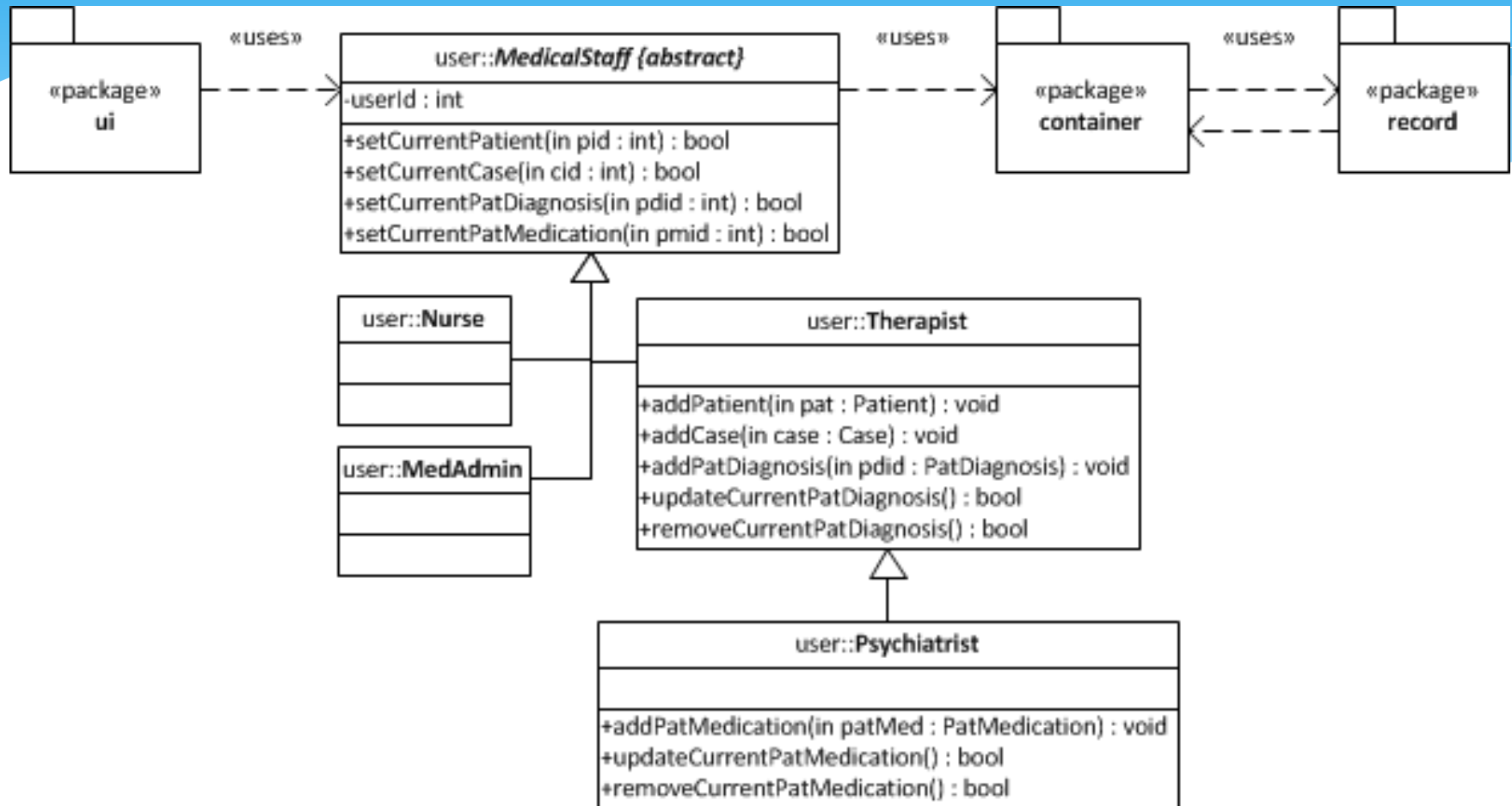


Packagestruktur

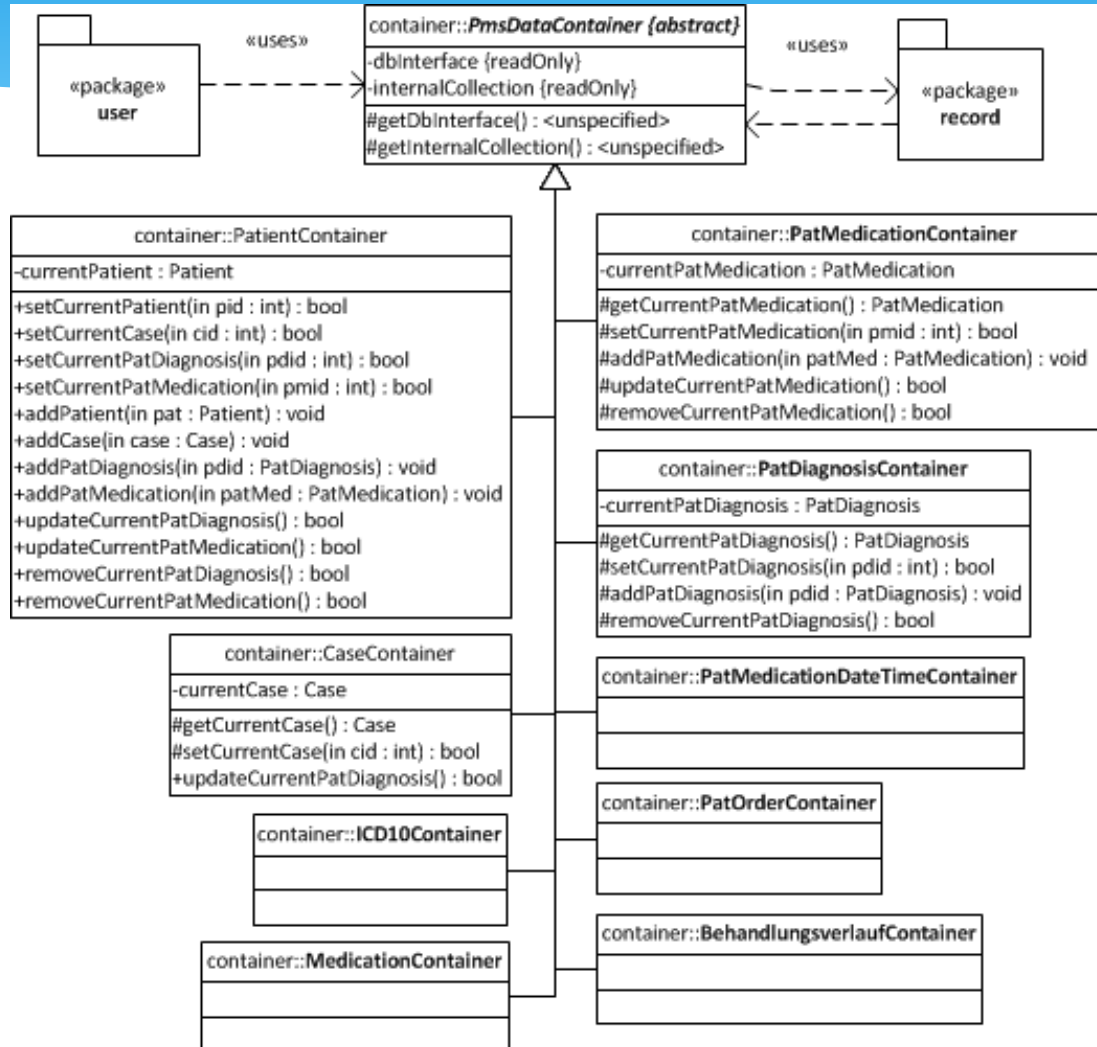
- * Container-Package: Kapselt eine Menge von Datensätzen eines bestimmten Typs (PatientContainer, PatDiagnosisContainer)



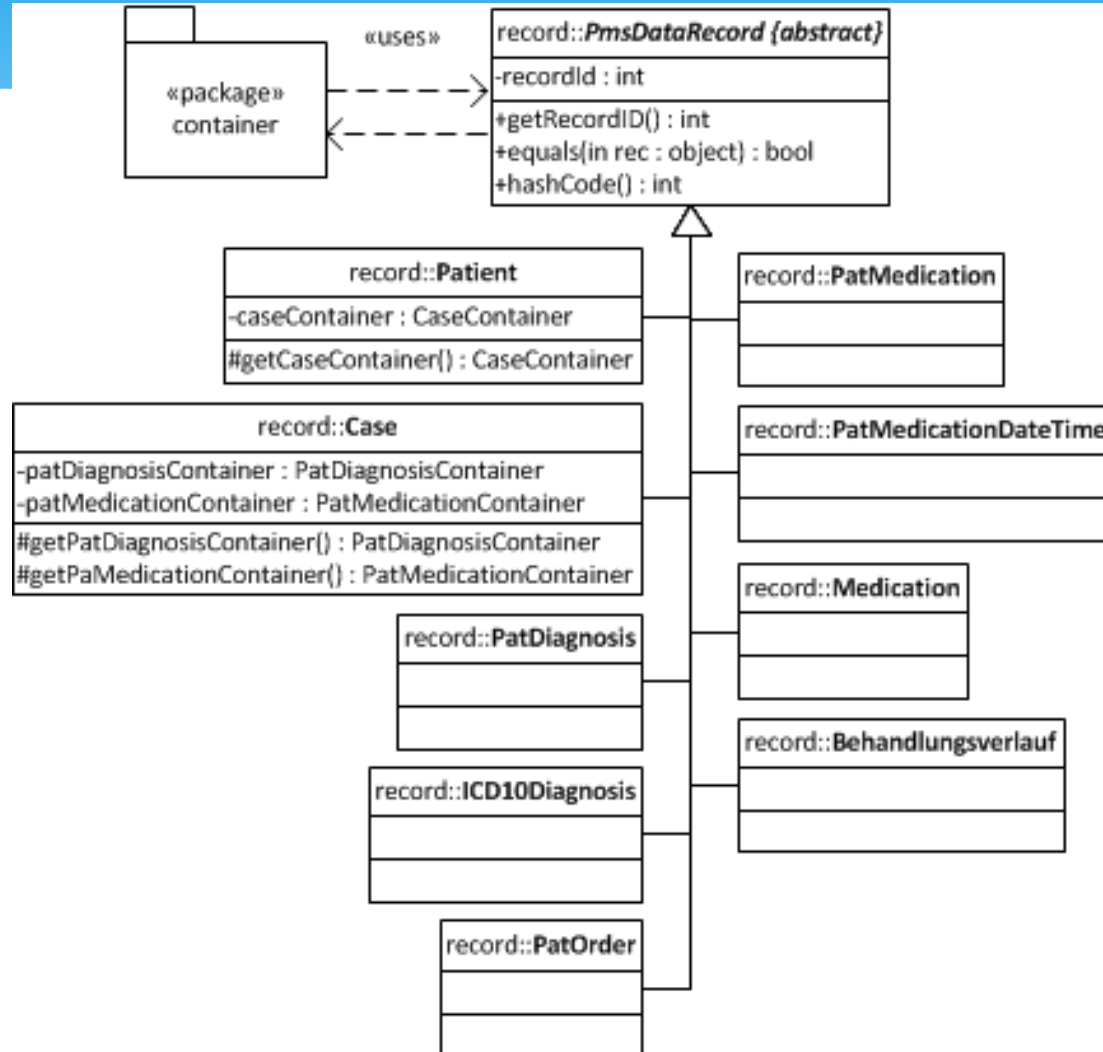
User-Klassen



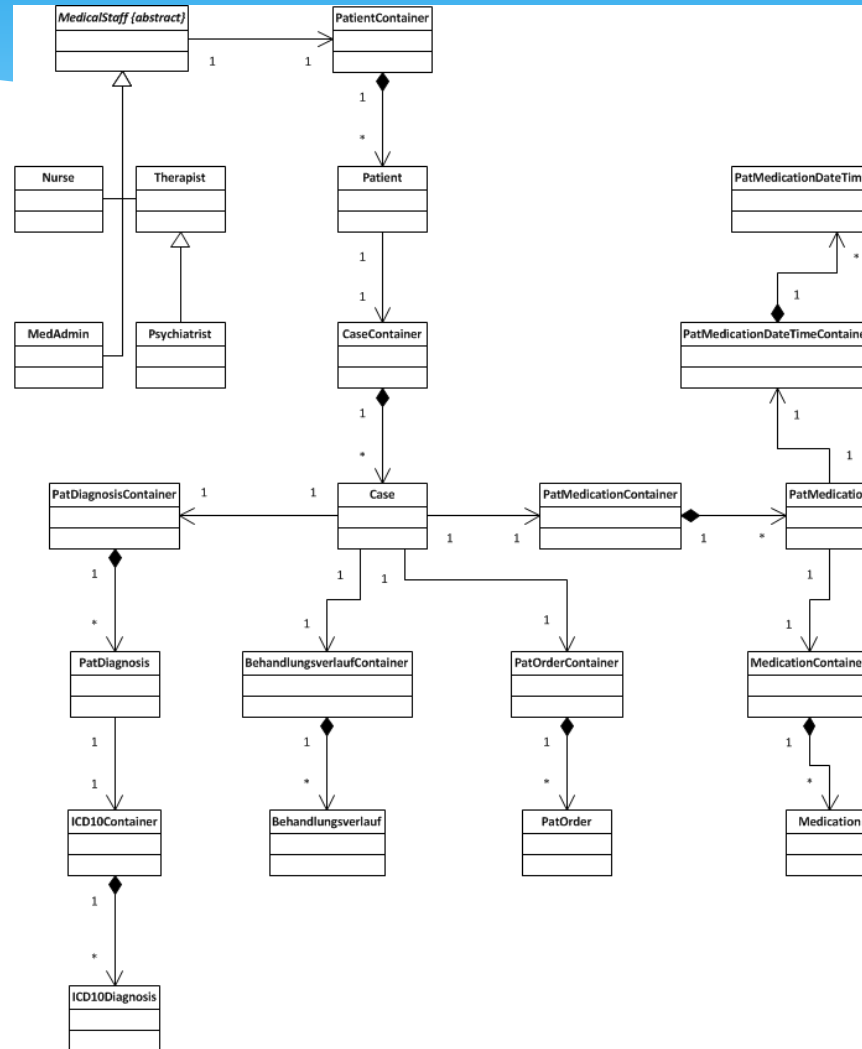
Container-Klassen



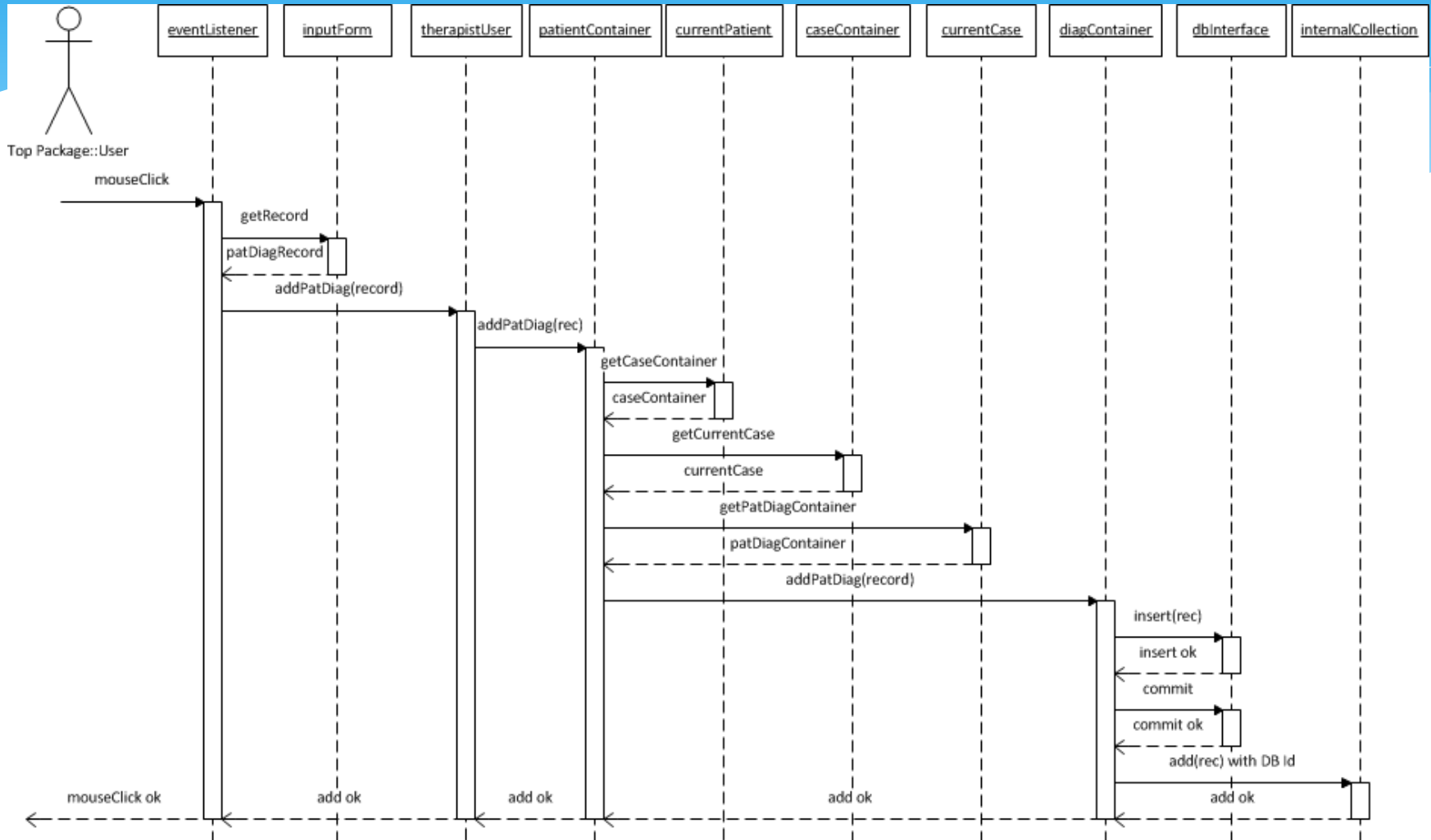
DataRecord-Klassen

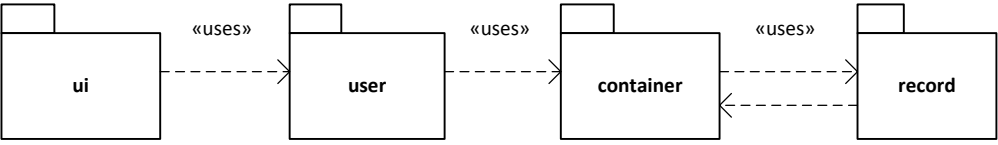


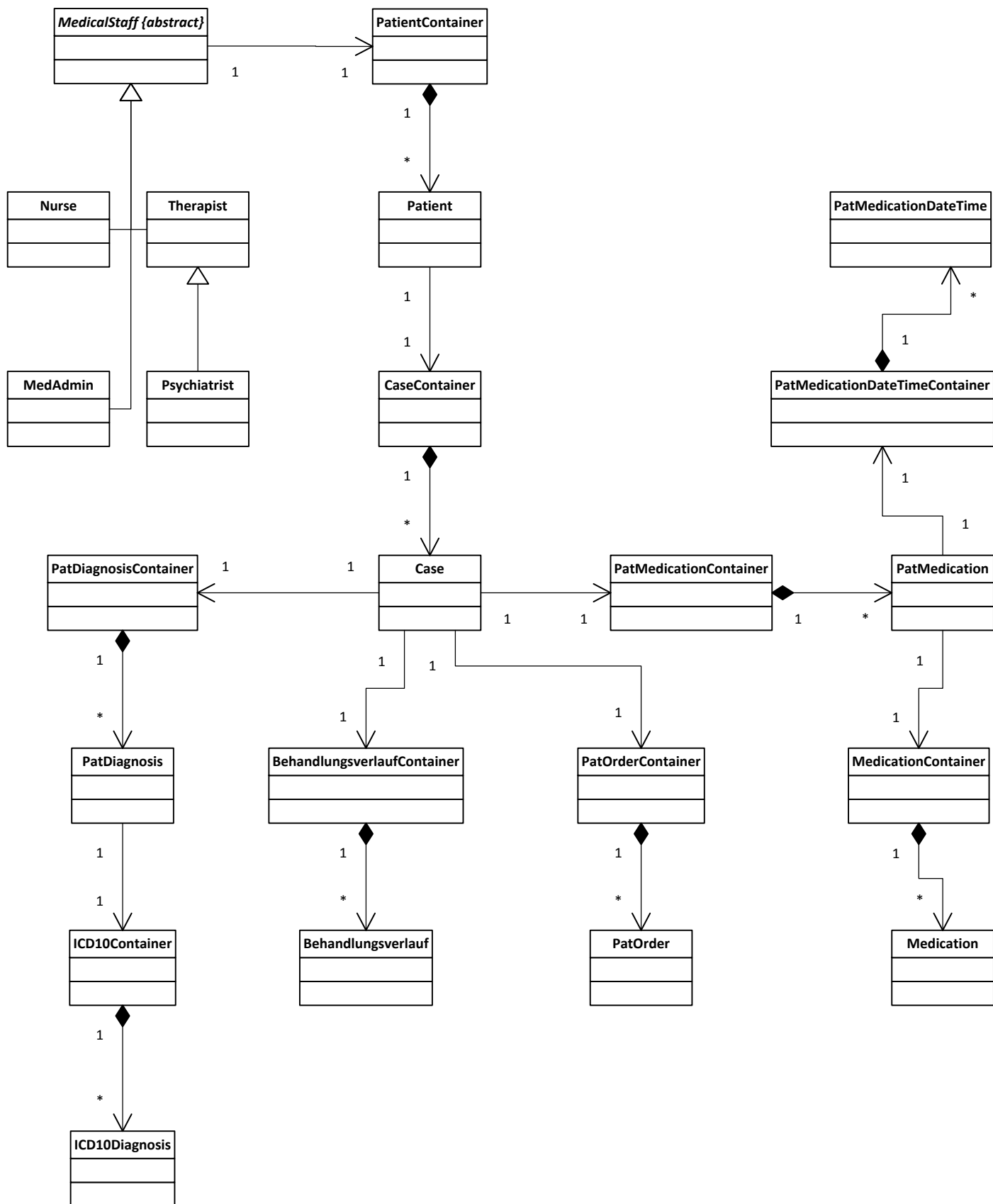
Aggregationen

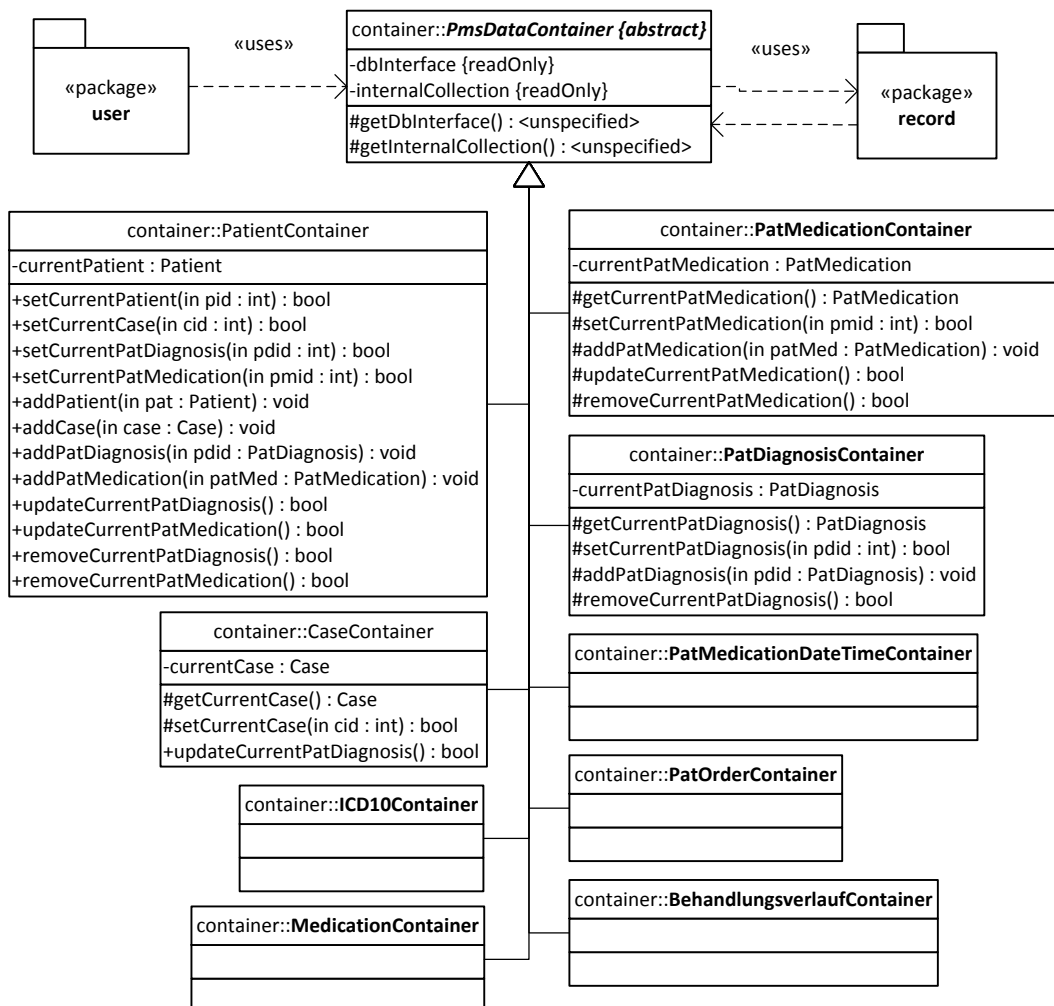
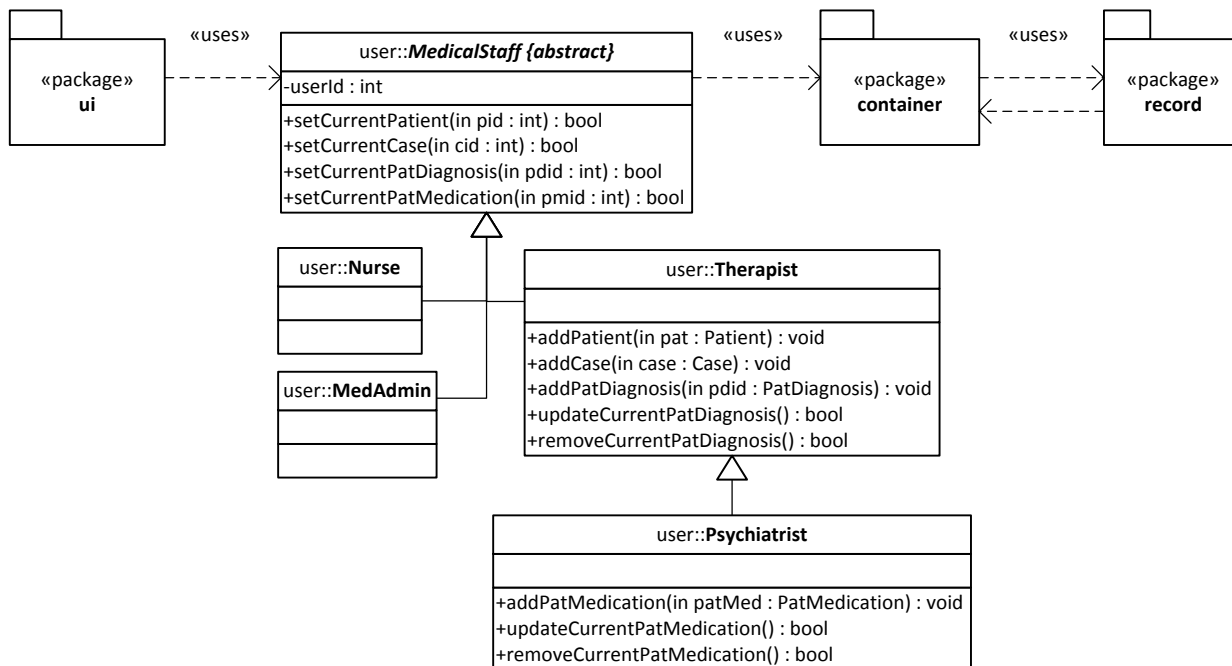
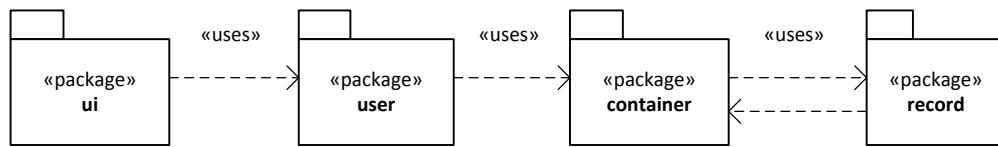


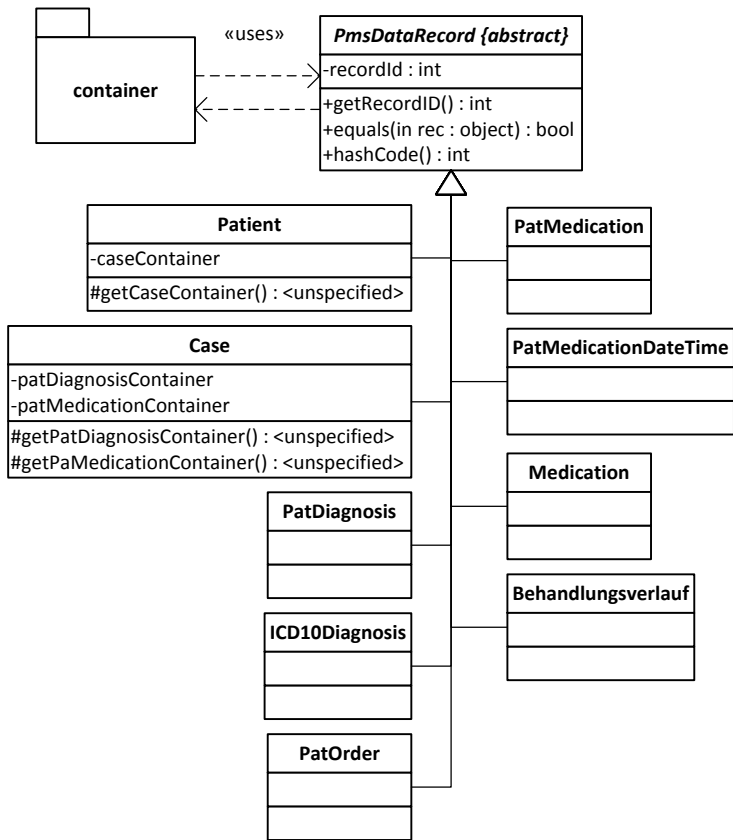
Sequenzdiagramm (neue Diagnose)

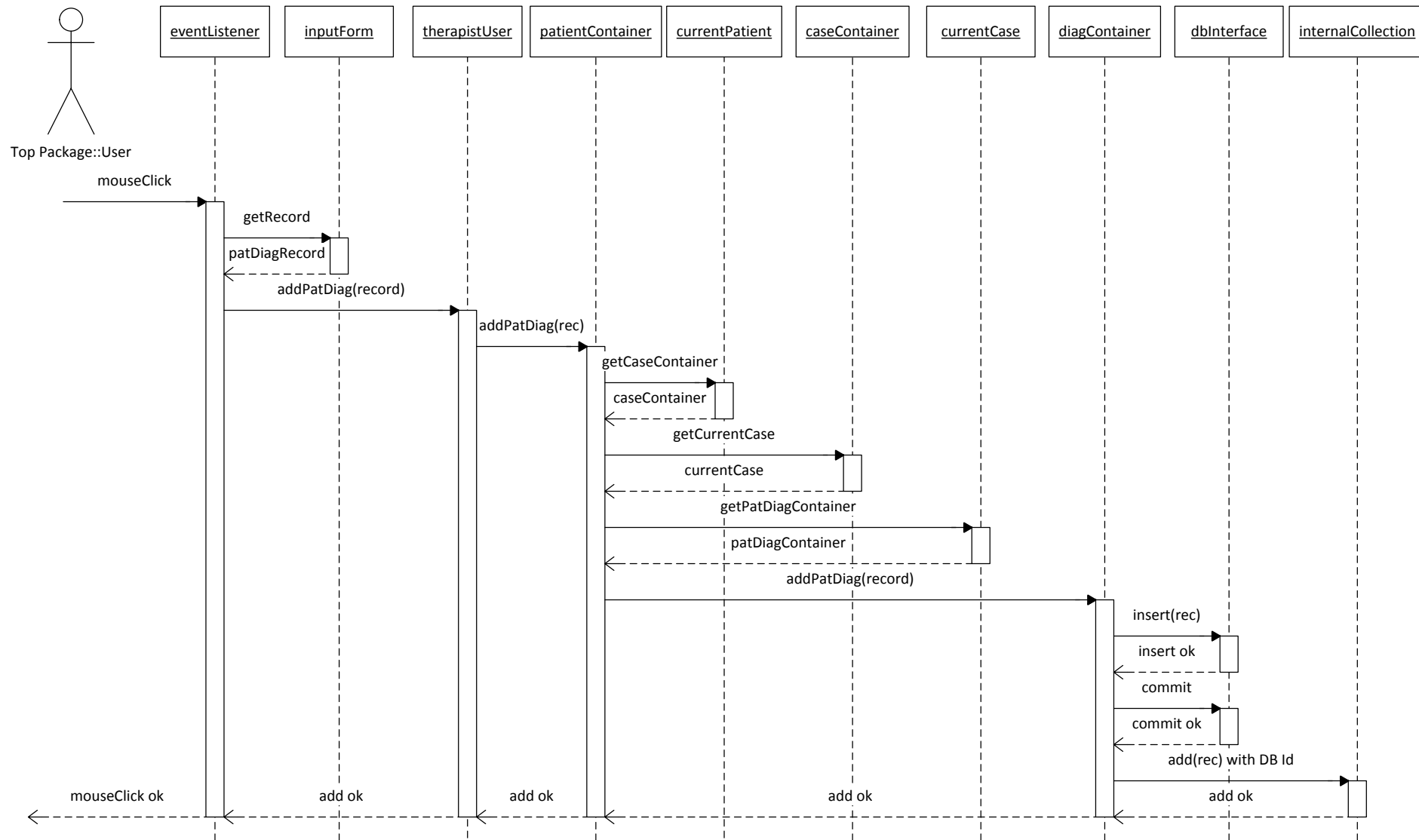












CRC-Karten

Arbeitsname:	MedicalStaff
Mögliche Namen:	MedicalStaff, STaffMedical, MedicalPersonnel
Effektiver Name:	MedicalStaff
Zweck:	Abstrakte Basisklasse zur Verwaltung der Zugriffsberechtigung und des aktuellen Zustandes eines Benutzers.

[illegible]



CRC-Karten

Arbeitsname:	Therapist
Mögliche Namen:	Therapist, StaffTherapist, TherapistStaff
Effektiver Name:	Therapist
Zweck:	Erweitert die Klasse MedicalStaff für den Benutzer Therapeut.

Verantwortlichkeiten	Kollaborationen
Erweiterte Zugriffsberechtigung auf die Methoden von PatientCollection	PatientCollection
neuen Patienten hinzufügen	PatientCollection
Neuen Fall hinzufügen	PatientCollection
Diagnose hinzufügen oder bearbeiten	PatientCollection
Auftrag hinzufügen oder bearbeiten	PatientCollection
Eintrag zum Krankheitsverlauf hinzufügen oder bearbeiten	PatientCollection

CRC-Karten

Arbeitsname:	Psychiatrist
Mögliche Namen:	Psychiatrist, StaffPsychiatrist, PsychiatristStaff
Effektiver Name:	Psychiatrist
Zweck:	Erweitert die Klasse Therapist für den Benutzer Psychiatrist.

[illegible]

CRC-Karten

Arbeitsname:	PatientContainer
Mögliche Namen:	PatientContainer
Effektiver Name:	PatientContainer
Zweck:	Diese Klasse verwaltet die vorhandenen Patienten-DataRecords.

[illegible]

CRC-Karten

Arbeitsname:	Case
Mögliche Namen:	Case, CaseSingle, SingleCase
Effektiver Name:	Case
Zweck:	DataRecord-Klasse für Verwaltung von fall-relevanten Daten

[illegible]



CRC-Karten

Arbeitsname:	PmsDataContainer
Mögliche Namen:	PmsDataContainer
Effektiver Name:	PmsDataContainer
Zweck:	Diese Klasse bietet einen gekapselten Zugriff auf eine Menge von DataRecords. Die abgeleiteten Klassen bestimmen, welche DataRecords eingefügt werden können.

Verantwortlichkeiten	Kollaborationen
Gekapselter Zugriff auf eine Menge von DataRecords (z. B. Patienten, Diagnosen, usw.) mit Einfüge- und Bearbeitungsoperationen.	andere Container- und DataRecord-Klassen
Verwaltung des Zustandes des Containers (z. B. currentRecord = aktueller Record des Containers)	interne Attribute des Containers (currentRecord)
Der Container delegiert die persistente Speicherung von DataRecords intern weiter (z. B. an die Implementierung eines DbInterfaces)	Objekt dbInterface
Ablage der DataRecords in einer nicht persistenten Datenstruktur, um Zugriff auf DB zu minimieren.	Objekt internalCollection



CRC-Karten

Arbeitsname:	PmsDataRecord
Mögliche Namen:	PmsDataRecord
Effektiver Name:	PmsDataRecord
Zweck:	Diese abstrakte Klasse kann erweitert werden, um den gekapselten Zugriff auf einen bestimmten DataRecord zu ermöglichen (z. B. Patient, Fall, Diagnose, ...).

Verantwortlichkeiten	Kollaborationen
Verwaltung einer eindeutigen Id zur Identifikation des Objektes.	Objekt recordId
getter- und setter-Methoden für die Datenfelder in den abgeleiteten Klassen	
Ein DataRecord kann einen Container enthalten (Abbildung einer one-to-many-Beziehung)	Container-Objekt