

UML Domain Model

- Ein **Caregiver** hat keine bis mehrere **Clients**.
- Ein **Client** hat ein (zuständiger) **Caregiver**.
- Ein **Caregiver** und ein **Client** haben je ein **Network**.
- Ein **Network** für einen **Client** kann keine bis mehrere **External** haben, enthält den **Caregiver**.
- Ein **Network** für einen **Caregiver** kann keine bis mehrere **External** haben.
- Ein **External** hat mehrere **Networks**.
- Ein **Appointment** gehört zu einem **Client** oder einem **Caregiver**.
- Ein **Appointment** gehört zu einem **Calendar**.
- Ein **Calendar** kann keine bis mehrere **Appointments** haben.
- Ein **Client** kann keine bis mehrere **Cases** haben.
- Ein **Case** gehört zu einem **Client**.
- Ein **Case** hat mind. einen bis mehrere **ActivityRecordings** (= Leistungen).
- Ein **ActivityRecording** ist in mind. einem bis in mehreren **Cases** erfasst sein.
- Eine **Bill** hat eine bis mehrere **ActivityRecordings**.
- Ein **ActivityRecording** zu einem bestimmten **Case**, ist auf mind. ein und max. auf zehn **Bill** gelistet.
- Von einem **Case** können mehrere **Statistics** erstellt werden.
- Eine **Statistics** kann mehrere **Cases** enthalten.

UML Sequence Diagram: User Update

Objekte: Caregiver, Client DB, Client, Case

- Caregiver sendet Clientanfrage an Client-Datenbank (query(clientName))
- ClientDB sendet eine Antwort (response())

Alternative 1: Antwort DB „Client nicht gefunden“

- Caregiver erstellt neuen Client (<<create client>>)
- Client-Objekt wird erstellt und sendet seine ID zurück (response(clientID))

Alternative 2: Antwort DB „Client gefunden“

- Caregiver erstellt neuen Fall für den Client (<<create case(clientID)>>)
- Fall-Objekt wird erstellt und sendet seine ID zurück (response(caseID))

UML Sequence Diagram: System Alert

Objekte: Caregiver, Client, Calendar, Appointment

- Caregiver ruft den Client auf (getClient(clientName))
- Client gibt seine ID zurück (response(clientID))
- Caregiver erstellt für den Client einen neuen Termin im Kalender, dafür gibt er neben der clientID auch das Datum und die Zeit von-bis mit (queryAppointment(date, period, clientID))
- Der Kalender überprüft, ob der Termin verfügbar ist (controlAppointment())

Alternative 1: Termin ist frei (response(true))

- Der Calendar erstellt den Termin (<<create>> Appointment)
- Der Appointment gibt eine Bestätigung an Calendar zurück (done())
- Der Calendar gibt eine Bestätigung an Caregiver zurück (done())

Alternative 2: Termin ist nicht verfügbar (response(false))

- Der Calendar gibt eine Fehlermeldung an Caregiver zurück (alert(appointmentNotAvailable))
- Der Caregiver stellt eine Anfrage an den Calendar, für einen freien Termin zu bestimmten Datum (queryAppointment(date))
- Der Calendar gibt einen Terminvorschlag zurück (response(appointmentSuggestion))
- Der Caregiver akzeptiert den Termin (accepted())
- Der Calendar erstellt den Termin (<<create>> Appointment)
- Der Appointment gibt eine Bestätigung an Calendar zurück (done())
- Der Calendar gibt eine Bestätigung an Caregiver zurück (done())