# Wichtige Designentscheidungen

# Apache Pivot (<http://pivot.apache.org/>)

Beim GUI Framework haben wir uns für Apache Pivot entschieden. Pivot ermöglicht die Konstruktion von visuell ansprechenden und plattformübergreifenden Anwendungen in Java. Außerdem ist Pivot komplett Open Source. Ein weiterer Grund für unsere Entscheidung war, dass Pivot eine gute Trennung von Logik und Darstellung erlaubt. Einfache Oberflächen können mit ziemlich einfach und schnell mit BXML erstellt werden. Ein gravierender Nachteil ist die Dokumentation. Die einzige Dokumentation, außer der Javadoc, sind die Tutorials auf der Homepage des Herstellers.

# Validierung

## Zeitraumvalidierung

Wenn bei einem Diätplan ein neuer Zeitraum hinzugefügt wird, wird validiert ob dieser Zeitraum mit der Zeitspanne einer anderen Diätbehandlung, der Zeitspanne eines anderen Diätplans und einem anderen Zeitraum im aktuellen Diätplan dieses Patienten kollidiert. Wenn eine Überschneidung auftritt wird in einer Infobox über den Zeiträumen angezeigt welche Zeiträume sich überschneiden. Denn ein Patient darf zu jeder Zeit nur einen Diätplan haben. Sprich: Ein Patient kann sich nicht an mehrere Diätpläne gleichzeitig halten und mehrere Mahlzeiten gleichzeitig essen.

## Parametervalidierung

Beim Anlegen von Diätbehandlungen müssen die zu erreichenden Zielparameter angegeben werden. Dies ist optional auch bei Diätplänen, Zeiträumen und Mahlzeiten möglich. Beim erstellen der Diätpläne mit EasyDiet wird der Diätassistent über die Einhaltung der festgelegten Zielparameter informiert. Dabei werden alle Parameter aller Mahlzeiten aller Zeiträume aller Diätpläne mit den Zielparametern verglichen.

# POJO

POJO ist die Abkürzung für (Plain Old Java Object) und ist somit ein „ganz normales“ Java Objekt. Hier werden alle Eigenschaften der Objekte festgelegt. Unser Domänenmodell ist besteht aus den POJOs. Wir bekommen die POJOs aus den Data Access Objekten (DAOs), die direkt über Hibernate von der Datenbank erstellt werden. Da POJOs nicht direkt an eine GUI übergeben werden sollten, verwenden wir dafür Businessobjekte(BOs).

# DAO

Bei Data Access Objekte handelt es sich um eine Programmierschnittstelle, mit der Programmierer auf Datenbanken zugreifen können. Die Daten aus der Datenbank können dann dargestellt und manipuliert werden. DAO ist als hierarchische Bibliothek mit allen Datenbank-Zugriffsrechten aufgebaut. Da wir im EasyDiet alle Persistenten Daten in einer Datenbank speichern, und mit Hibernate auf diese zugreifen, verwenden auch wir Data Access Objects. Aus den DAOs aus der Datenbank bekommen wir die POJOs (Plain Old Java Objects). Die Basisklasse implementiert die Allgemeinen Funktionalitäten, in Unterklassen sind die Spetialisierungen implementiert.

# BO

Da man POJOs (Plain Old Java Objects) nicht direkt auf eine GUI übergeben sollte, benötigt man eine Schnittstelle zwischen POJOs und GUI. Businessobjekte wurden als diese Schnittstelle eingeführt. Die Businessobjects (BOs) werden also zur Weitergabe der POJOs an die GUI genutzt. Die BOs kapseln die für den Anwender gebrauchten Daten.