# Wichtige Designentscheidungen

# Apache Pivot

Wir haben uns für das GUI Framework Apache Pivot entschieden. Pivot ermöglicht die Konstruktion von visuell ansprechenden und plattformübergreifenden Anwendungen in Java. Neben dem Grund, dass Pivot komplett Open Source ist, war ein weiterer Grund für unsere Entscheidung, dass Pivot eine optimale Trennung von Logik und Darstellung erlaubt. Einfache Oberflächen können einfach und schnell durch XML Dokumenten definiert werden. Ein gravierender Nachteil ist die Dokumentation welche in der aktuellen Version 2.0 nur aus der Javadoc und den Tutorials auf der Webseite des Herstellers besteht. Der Quellcode selbst ist jedoch auch eine gute Anlaufstelle um das Verhalten von Pivot ergründen zu können.

# Validierung

## Zeitraumvalidierung

Wenn bei Diätplanen die Zeiträume manipuliert werden, wird validiert ob diese mit anderen Zeitspannen desselben Diätplans, mit einem anderen Diätplan oder einer anderen Diätbehandlung zeitlich kollidiert. Wenn eine Überschneidung auftritt wird in einer Infobox über den Zeiträumen angezeigt welche Zeiträume sich überschneiden um den Benutzer auf seinen Fehler aufmerksam zu machen. Es gilt global die Regel: Ein Patient darf zu jeder Zeit nur einen zugeteilten Zeitraum haben. Dies stellt sicher, dass der Patient nur ein sog. Mahlzeitenset zur Auswahl hat.

## Parametervalidierung

Beim Anlegen von neuen Diätplänen müssen die zu erreichenden Zielparameter angegeben werden. Beim Erstellen der Diätpläne mit EasyDiet wird der Diätassistent über die Einhaltung der festgelegten Zielparameter informiert. Die Parameter der einzelnen Rezepte die zum Diätplan hinzugefügt werden, werden aufsummiert und es wird geprüft ob man noch im zuvor festgelegten Bereich ist. Parameter können in tieferen Schichten überschrieben werden, sodass in diesen Schichten die Parameter der höheren Schicht ignoriert, und stattdessen die eigenen verwendet werden. Bsp: In Mahlzeiten wird der gültige Bereich des Brennwertes vom Diätplan überschrieben.

# 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 und sonst keine Logik implementiert. Die POJOs bilden die relationale Datenbank als objektorientierte Datenstruktur mit Assoziationen ab. Unser Domänenmodell bzw. Datenmodell besteht aus solchen POJOs welche dynamisch aus der Hibernate Konfiguration generiert wurden. POJOs werden nicht über die Domänenschicht hinaus weitergegeben sondern davor in Business-Objekte gekapselt. Über Data Access Objects werden POJOs im DataStore (MySQL Datenbank) persistiert und geladen.

# 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 gekapselt und in objektorientierter Form verwendet werden. Die DAO Implementationen von EasyDiet verwenden intern die Hibernate Bibliothek um mit der Datenbank zu kommunizieren. Sie kapseln somit die Datenbankanbindung nach außen als Schnittstelle. Aus den DAOs aus der Datenbank bekommen wir die POJOs (Plain Old Java Objects). Die Basisklasse implementiert lediglich die Eigenschaften der Datenbank und der relationalen Assoziationen von Tabellen.

# BO

Business-Objekte (Bos) kapseln POJOs um diese in ihren Basisfunktionalitäten zu erweitern. Die Business-Objekte stellen somit eine Datenintegrität und Datenvalidität zur Verfügung. BOs kapseln alle für die GUI relevanten Eigenschaften und Funktionalitäten weshalb diese auch als solche an die Benutzeroberfläche weitergegeben werden. Durch die Weitergabe der speziell für die Benutzeroberfläche konzipierten Objekte an Apache Pivot können wir spezielle Features wie DataBindung und GUI Technische Fehlerbehandlungen nutzen. Diese Business-Objekte werden nicht weiter in Interfaces oder Delegations-Controller gekapselt, da die angebotenen Funktionalitäten zwar GUI Framework unabhängig, aber doch speziell für Oberflächen implementiert wurden.