|  |
| --- |
| A-Team |
| EasyDiet |
| Pflichtenheft |

|  |
| --- |
| Daniel Kuschny;Simon Moosbrugger;Mathias Joschika;Matthias Amann  25.03.2011 |

EasyDiet  
Pflichtenheft (SRS - Software Requirements Specification)   
Version <1.1>

# Revisionsverlauf

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Beschreibung | Author |
| 18.03.2011 | 1.0 | Einfügen Stakeholder, Einführung-System | Simon Moosbrugger |
| 21.03.2011 | 1.1 | Einführung, Stakeholder und Teil der Produkt Überblick | Mathias Joschika |
| 25.03.2011 | 1.2 | Übernahme von Google-Docs, Neuformattierung, Zusammenführen aller bisherigen Arbeiten | Daniel Kuschny |

Inhaltsverzeichnis

[1 Revisionsverlauf 1](#_Toc288834225)

[2 Einführung 7](#_Toc288834226)

[2.1 System 7](#_Toc288834227)

[2.2 Zweck 7](#_Toc288834228)

[2.3 Umfang 7](#_Toc288834229)

[2.4 Referenzen 7](#_Toc288834230)

[2.5 Abkürzungen 7](#_Toc288834231)

[2.6 Überblick 8](#_Toc288834232)

[3 Stakeholder- und Benutzerbeschreibung 9](#_Toc288834233)

[3.1 Stakeholder/Benutzer Zusammenfassung 9](#_Toc288834234)

[3.2 Benutzerumgebung 11](#_Toc288834235)

[4 Produkt Überblick 12](#_Toc288834236)

[4.1 Zusammenfassung Produktfähigkeiten/Eigenschaften 12](#_Toc288834237)

[4.2 Produkt Fähigkeiten/Eigenschaften 13](#_Toc288834238)

[4.2.1 Systembenutzerverwaltung 13](#_Toc288834239)

[4.2.2 Patientenstammverwaltung 13](#_Toc288834240)

[4.2.3 Diätplan erstellen 13](#_Toc288834241)

[4.2.4 Rezepte anlegen 13](#_Toc288834242)

[4.2.5 Patientenstatus anlegen 14](#_Toc288834243)

[4.2.6 Ernährungsprotokoll anlegen 14](#_Toc288834244)

[4.2.7 Diätplanvorlage erstellen 14](#_Toc288834245)

[4.2.8 Diätvorlage erstellen 14](#_Toc288834246)

[4.2.9 Parameterset anlegen 14](#_Toc288834247)

[4.2.10 Diätbehandlung anlegen 14](#_Toc288834248)

[4.3 Annahmen und Abhängigkeiten 14](#_Toc288834249)

[5 Domänenmodell 15](#_Toc288834250)

[5.1 Überblick 15](#_Toc288834251)

[5.2 Detailliertes Modell 16](#_Toc288834252)

[5.2.1 Patient 16](#_Toc288834253)

[5.2.2 FamilienAnamnese 16](#_Toc288834254)

[5.2.3 Diätbehandlung 17](#_Toc288834255)

[5.2.4 Behandlungsvorlage 17](#_Toc288834256)

[5.2.5 Behandlungsstatus 18](#_Toc288834257)

[5.2.6 Patientenstatus 19](#_Toc288834258)

[5.2.7 PatientenstatusTyp 19](#_Toc288834259)

[5.2.8 Laborbefund 19](#_Toc288834260)

[5.2.9 Kontaktjournal 20](#_Toc288834261)

[5.2.10 KontaktTyp 20](#_Toc288834262)

[5.2.11 Ernährungsprotokoll 20](#_Toc288834263)

[5.2.12 ErnährungsprotokollTyp 20](#_Toc288834264)

[5.2.13 Plan 21](#_Toc288834265)

[5.2.14 PlanTyp 21](#_Toc288834266)

[5.2.15 Zeitraum 22](#_Toc288834267)

[5.2.16 Mahlzeit 22](#_Toc288834268)

[5.2.17 MahlzeitenZeile 22](#_Toc288834269)

[5.2.18 Rezept / Nahrungsmittel 23](#_Toc288834270)

[5.2.19 Nahrungsmittelparameter 23](#_Toc288834271)

[5.2.20 Parameterdefinition 24](#_Toc288834272)

[5.2.21 Parameterset 24](#_Toc288834273)

[5.2.22 DiätparameterTyp 24](#_Toc288834274)

[5.2.23 Diätparameter 25](#_Toc288834275)

[5.2.24 DiätparameterSet 25](#_Toc288834276)

[5.2.25 Systembenutzer 26](#_Toc288834277)

[5.2.26 Rolle 26](#_Toc288834278)

[5.3 Einschränkungen (Constraints) 27](#_Toc288834279)

[5.3.1 Patient 🡪 Diätbehandlung 27](#_Toc288834280)

[5.3.2 Diätbehandlung 🡪Plan 27](#_Toc288834281)

[5.3.3 Mahlzeit 🡪 Diätparameter / Mahlzeitenzeile 27](#_Toc288834282)

[6 Dynamisches Modell 28](#_Toc288834283)

[6.1 Detaillierte Benutzungsfälle (UseCases) 28](#_Toc288834284)

[6.1.1 Systembenutzer anlegen 28](#_Toc288834285)

[6.1.2 Patientenstamm anlegen 28](#_Toc288834286)

[6.1.3 Diät anlegen 29](#_Toc288834287)

[6.1.4 Ernährungsprotokoll anlegen 31](#_Toc288834288)

[6.1.5 Rezepte anlegen 34](#_Toc288834289)

[6.1.6 Patientenstatus anlegen 36](#_Toc288834290)

[6.1.7 Laborbefund anlegen 36](#_Toc288834291)

[6.1.8 Diätplan anlegen 36](#_Toc288834292)

[6.1.9 Parameterset anlegen 39](#_Toc288834293)

[6.1.10 Diät im Diätpool anlegen 40](#_Toc288834294)

[6.1.11 Diätplan im Diätpool anlegen 41](#_Toc288834295)

[6.2 Objekt Lifecycles 43](#_Toc288834296)

[6.2.1 Lebenszyklus einer Diätbehandlung 43](#_Toc288834297)

[7 Nichtfunktionale Anforderungen 44](#_Toc288834298)

[7.1 Regeln 44](#_Toc288834299)

[7.2 Usability 44](#_Toc288834300)

[7.3 Zuverlässigkeit 44](#_Toc288834301)

[7.4 Performanz 44](#_Toc288834302)

[7.5 Unterstützbarkeit 44](#_Toc288834303)

[7.6 Online Benutzerdokumentation und Help System 45](#_Toc288834304)

[7.7 Zugekaufte Komponenten 45](#_Toc288834305)

[7.8 Schnittstellen 45](#_Toc288834306)

[7.8.1 Benutzerschnittstellen 45](#_Toc288834307)

[7.8.2 Software Schnittstellen 45](#_Toc288834308)

[7.8.3 Kommunikationsschnittstellen 45](#_Toc288834309)

[7.9 Zusätzliche Lizenzierungen 45](#_Toc288834310)

[7.10 Copyright und andere rechtliche Anforderungen 45](#_Toc288834311)

[7.11 Anzuwendende Standards 46](#_Toc288834312)

[8 Iterationenplan (Timeboxes) 47](#_Toc288834313)

[8.1 Überblick 47](#_Toc288834314)

[8.2 Timebox 1 48](#_Toc288834315)

[8.2.1 Benutzungsfall/fälle (UseCase(s)) 48](#_Toc288834316)

[8.2.2 Architektur 48](#_Toc288834317)

[8.2.3 Deliverables 48](#_Toc288834318)

[8.2.4 Abhängigkeiten 48](#_Toc288834319)

[8.3 Timebox 2 49](#_Toc288834320)

[8.3.1 Benutzungsfall/fälle (UseCase(s)) 49](#_Toc288834321)

[8.3.2 Architektur 49](#_Toc288834322)

[8.3.3 Deliverables 49](#_Toc288834323)

[8.3.4 Abhängigkeiten 49](#_Toc288834324)

[8.4 Timebox 3 50](#_Toc288834325)

[8.4.1 Benutzungsfall/fälle (UseCase(s)) 50](#_Toc288834326)

[8.4.2 Architektur 50](#_Toc288834327)

[8.4.3 Deliverables 50](#_Toc288834328)

[8.4.4 Abhängigkeiten 50](#_Toc288834329)

[9 Glossar 51](#_Toc288834330)

# Einführung

## System

Das System wird von DA’s zur Unterstützung bei der Erstellung von Diätplänen für Patienten im AKH Wien verwendet. Es offeriert nicht nur diese Funktion sondern bietet zusätzlich noch die Möglichkeit EP’s und Laborbefunde zu erfassen, Statistiken anhand dieser zu erzeugen, Rezepte zur Verfügung zu stellen und bietet die Möglichkeit alle Ereignisse die im Zusammenhang mit dem Patienten stehen zu protokollieren.

## Zweck

Der Zweck des Pflichtenheftes/ SRS ist die detaillierte und strukturierte Analyse und Beschreibung der Kundenanforderungen, um die Vollständigkeit des zu entwickelnden Systems sicherzustellen.

Die Zielgruppe ist der Kunde (Definition der Anforderungen), der zuständige technische Projektleiter (Analyse und Dokumentation der Anforderungen), und die Qualitätssicherung (Review)

## Umfang

Dieses Dokument ist nur für das Projekt EasyDiet gültig.

Der Umfang des Projektes umfasst:

* Das Produkt soll als Java-Applikation sowie als Java-Applet angeboten werden.
* Durch die Verwendung von Java ist das Produkt Plattformunabhängig.

Dieses Projekt ist vollkommen unabhängig von anderen Projekten und daher als eigenständiges Projekt zu sehen.

## Referenzen

* Requirements-Workshop
* Systembeschreibung
* Bundeslebensmittelschlüssel

## Abkürzungen

Definition und Beschreibung aller in diesem Dokument verwendeten Abkürzungen.

|  |  |
| --- | --- |
| Abkürzung | Beschreibung |
| DA | Der/die DiätassistentIn |
| BLS | Bundeslebensmittelschlüssel |
| EP | Ernährungsprotkoll |

## Überblick

Das Dokument ist in zwei Teile gegliedert:

* technisches Wissen erforderlich
* technisches Wissen nicht erforderlich

Nur bei Punkt 5 ist technisches Wissen erforderlich und daher nicht für unerfahrene Leser geeignet. In Punkt 5 wird das Domänenmodell besprochen, bei dem Design und Analytisches Wissen vorausgesetzt wird. Bei den anderen Punkten muss kein technisches Wissen vorhanden sein.

# Stakeholder- und Benutzerbeschreibung

Das Erstellen eines Diätplanes birgt viele Möglichkeiten Fehler zu machen. Fehler wie:

* Vorlieben und Abneigungen vergessen oder vertauschen
* Allergien übersehen
* keine Kenntnis über die exakten Inhaltsstoffe eines Rezeptes

Da diese Fehler die Motivation und die Zufriedenheit des Patienten erheblich senken, ist es wichtig, dass diese so gut wie möglich vermieden werden. Durch die Unterstützung des Systems ist die Minimierung dieser Fehler möglich

Ebenso bei der Patientenverwaltung können wichtige Patientendaten vergessen oder vertauscht werden. Wenn dies der Fall ist, muss der Patient extra kontaktiert werden und dies beansprucht extra Zeit und Aufwand.

## Stakeholder/Benutzer Zusammenfassung

| Name | Rolle/Funktion | interessiert an |
| --- | --- | --- |
| Arzt/Ärztin | erstellt die Zuweisungsdiagnose und leitet den Patienten an den Diatässistenten weiter | schnellen Genesung des Patienten,  dass der DA seine Vorgaben umsetzt, |
| DiätassistentIn | Erstellung von Diätplänen, Erhebung des Patientenstatus, entscheidet und überwacht die Diät | dass der Patient die Vorgaben einhält und dadurch den größtmöglichen Nutzen aus Diätbehandlung erhält, an einer guten Unterstützung bei der Diätplanerstellung durch das System, an der Auswertungen der verschiedenen Pläne und Protokolle |
| DA-AssistentIn (SekretärIn, PraktikantIn, StudentIn) | eintragen von EP’s;  Stammdatenpflege; Rezeptverwaltung;  Drucken von Plänen, Vorlagen, etc | einer schnellen Eingabe von EP’s,  der einfachen Stammdatenpflege,  einer unkomplizierten Rezeptverwaltung |
| PatientIn | bekommt eine Diät verschrieben | einer möglichst effizienten Behandlung,  dass auf seine Vorlieben und Abneigungen eingegangen wird, |
| Koch/Köchin | kocht nach den Diätplänen die die DA’s erstellt haben | übersichtliche Darstellung der Rezepte und genaue Auflistung der Zutaten |
| IT-MitarbeiterIn | Wartungsarbeit | einem stablien System,  einer einfachen Wartbarkeit dessen,  einfache Integration in vorhandene Systeme |
| KrankenpflegerIn | eintragen von EP’s | der einfachen Eingabe von Ernährungsprotokollen |
| VersicherungsmitarbeiterIn | Ansehen von Statistiken | einem einfachen Zugang und übersichtlicher Darstellung der Daten |
| Krankenkasse | Bezahlung der anfallenden Behandlungskosten | einer exakten Auflistung der in Anspruch genommenen Leistungen, |
| SoftwareentwicklerIn/TesterIn | Weiterentwicklung des Systemes | der Funktionalität des Programmes und der Weiterentwicklung des Systems |
| Bundesgesundheitsamt | liefert BLS | der Verwendung ihres BLS zur Sicherstellung der Vereinheitlichung |
| Systeminhaber (Stadt Wien) | Finanziert die Hard- und Software | möglichst niedrigen Projektkosten und kurzer -dauer |

## Benutzerumgebung

Durchschnittlich sind bei einem gesamten Durchlauf aller Usecases 2 verschiedene Personen integriert. Als Erstes die Sekretärin, welche die Stammdaten des Patienten in das dazugehörige Formular einträgt.

Als zweites der DA, der die Diätpläne zusammenstellt. Diese 2 sind fixiert und keine anderen Personen bedienen das System. Die Dauer eines solchen Durchlaufes variiert von wenigen Minuten bis hin zu mehreren Wochen, je nachdem welcher Usecase durchgeführt wird. Die Abläufe innerhalb dieses Durchlaufes haben jedoch konstante Zeitaufwände.

Bis jetzt besitzt jeder DA einen PC auf dem Windows XP installiert ist. Jeder PC ist mit dem Netzwerk des AKH Wien verbunden. Bisher wurde zur Erfassung der Daten nur Microsoft Word und Excel verwendet. Es wird darum gebeten, dass auch die zukünftige Applikation auf dieser Architektur laufen muss.

# Produkt Überblick

Das Programm EasyDiet soll DA’s bei ihrer Arbeit unterstützen. Es können die Stammdaten des Patienten eingepflegt werden und es kann im Kontaktjournal mitprotokolliert werden.

Das Hauptfeature des Systems ist das Erstellen von Diätplänen. Hier kann der/die DA für den jeweiligen Patienten aus einer vorgefilterten Liste Rezepte und Nahrungsmittel für die verschiedenen Mahlzeiten auswählen.

Im Nachhinein können diverse Auswertungen gemacht werden. Grundlage für diese Auswertungen sind die EP’s, die Diätparameter des Diätplanes und die Laborbefunde.

Im Krankenhaus befindet sich bereits die benötigte Hardware, da die DA’s bisher den Computer aber nur zum Nachsehen von Patientendaten verwendet haben, soll nun ein unterstützendes Programm eingeführt werden.

## Zusammenfassung Produktfähigkeiten/Eigenschaften

| Produktfähigkeit | Stakeholder Nutzen/Gewinn |
| --- | --- |
| Diätplan erstellen | DA: Die Rezepte können nach Abneigungen und Allergien des Patienten gefiltert werden. Außerdem kann der DA Zielparameter hinzufügen, welche er später bei der Soll-Ist Analyse vergleichen kann. Durch das Hinzufügen der verschiedenen Rezepte wird automatisch berechnet, ob sich die aktuellen Werte noch im Bereich der angegebenen Zielparameter oder diese bereits überschritten haben. |
| Soll - Ist Vergleich | DA: Durch den Soll - Ist Vergleich kann der DA die Entwicklung des Patienten anhand der Zielparameter verfolgen. Falls die Werte zu stark abweichen, kann der DA reagieren und Maßnahmen ergreifen. Ebenso kann kontrolliert werden ob der DA bis jetzt die richtige Behandlungsform gewählt hat. |
| Patientenverwaltung | DA: Durch die Patientenverwaltung können Stammdaten sowie medizinische Daten eines Patienten zusammen zentral abgelegt werden und dort abgefragt werden. Der Patient kann eindeutig identifiziert werden. Der Arzt sowie der DA haben Zugang zu den Daten und es müssen nicht Daten mehrfach eingegeben werden und gar redundant gespeichert werden. |
| Rezeptverwaltung | DA: Durch die Rezeptverwaltung kann der DA die verschiedenen Rezepte immer wieder verwenden, ohne sie jedes Mal anlegen zu müssen. Nicht nur die Wiederverwendbarkeit, sondern auch das einfache Hinzufügen stellt sicher, dass die Rezeptdatenbank immer auf dem aktuellsten Stand ist. |
| Patientenstatus | DA: Durch den Patientenstatus kann der DA genau feststellen wie die momentane psychische sowie physische Verfassung des Patienten ist. Dank dieser Information kann der Da nicht nur die Diätpläne genau auf die Bedürfnisse des Patienten anpassen sondern weiß auch wie er dem Patienten am besten umgehen soll. |
| Diätplanvorlage anlegen | DA: Da Diätpläne oft ähnlich sind, kann der DA Diätplanvorlagen erstellen. Wenn nun ein neuer Diätplan erstellt wird, muss er nur noch die sich ändernden Daten eingeben und erspart sich somit Zeit und Aufwand. |

## Produkt Fähigkeiten/Eigenschaften

### Systembenutzerverwaltung

Es können Systembenutzer im System angelegt bearbeitet und gelöscht werden. Nach dem Anlegen können die Systembenutzer sich am System anmelden und die Aktionen durchführen zu denen sie die Berechtigung haben.

### Patientenstammverwaltung

Das System EasyDiet unterstützt eine Patientenstammdatenverwaltung in der Patienten erfasst, geändert und gelöscht werden können. Bei der Erfassung eines neuen Patientenstammes wird ein Formular geöffnet, bei dem alle erforderlichen Daten eingegeben werden können. Die hier angelegten Daten können später von den Systembenutzern eingesehen und dementsprechend verwendet werden

### Diätplan erstellen

Der Diätplan beinhaltet eine Auflistung der zu essenden Rezepte welche vom DA für den Patienten ausgewählt wurden. Der Bindungsgrad dieser Angaben hängt vom Typ des Diätplanes ab. Der Diätplan wird dem Patienten ausgehändigt und dieser sollte sich so gut wie möglich an die Vorgaben halten. Der DA kann dann anhand des Diätplanes und der Ernährungsprotokolle einen Soll-Ist Vergleich erstellen.

### Rezepte anlegen

Rezepte können angelegt, geändert und gelöscht werden. Nach dem Anlegen eines Rezeptes kann es zu mehreren Diätplänen hinzugefügt werden.

### Patientenstatus anlegen

EasyDiet ermöglicht es für jeden Patienten mehrere Patientenstatus anzulegen welcher die momentane psychische und physische Verfassung des Patienten wiederspiegelt. Dieser Status teilt sich auf in Laborbefunde sowie Einschätzungen des DA.

### Ernährungsprotokoll anlegen

Das Ernährungsprotokoll ist eine Auflistung von Mahlzeiten. Es wird aber die tatsächliche Menge die der Patient verzehrt angegeben und daher dient das Ernährungsprotokoll als solide Grundlage für Auswertungen.

### Diätplanvorlage erstellen

Der DA kann mit Hilfe der Diätplanvorlagen standardisierte Diätpläne erstellen. Wenn viele Patienten einen ähnlichen Plan haben, kann er einen allgemeinen Diätplan erstellen, bei dem er nur kleine Veränderungen vornehmen muss.

### Diätvorlage erstellen

Das Erstellen einer Diätvorlage ist identisch mit dem Erstellen einer Diät. Der Hintergedanke hierbei ist, dass es standardisierte Diäten gibt, die für alle Menschen gleich sind.

### Parameterset anlegen

Ein Parameterset dient zur Gruppierung von Parametern. Parametersets können ganz einfach an Diätpläne angehängt werden und erhöht dadurch die Wiederverwendbarkeit.

### Diätbehandlung anlegen

Im EasyDiet System ist es möglich für einen Patient unabhängige Diätbehandlungen anzulegen. Die Diät ist das zentrale Element bei der Durchführung einer Diätbehandlung. Es können verschiedenste Protokolle und Pläne angehängt werden.

## Annahmen und Abhängigkeiten

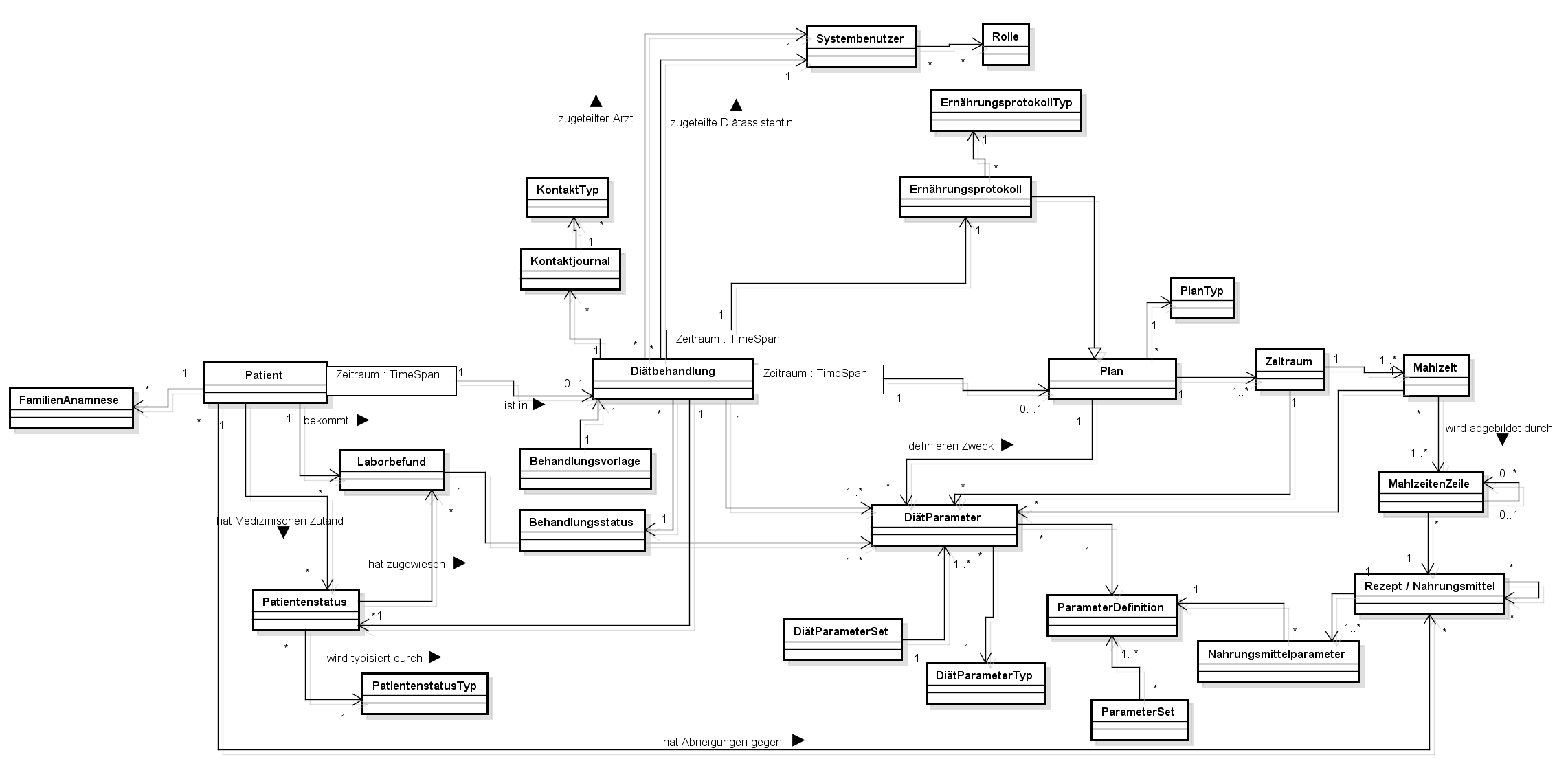
Auf jedem PC auf dem EasyDiet ausgeführt wird muss eine - entsprechend für das Betriebssystem passende – Java Virtual Machine installiert sein.

Der Datenbankserver muss zu jederzeit verfügbar sein. Außerdem muss das Betriebssystem grafische Benutzeroberflächen unterstützen.

# Domänenmodell

Das Domänenmodell gibt einen Überblick über die im System vorhandenen Elemente und deren Beziehung miteinander.

## Überblick



## Detailliertes Modell

In den folgenden Sektionen werden die einzelnen Entitätstypen des Domänenmodells genauer erläutert und deren Aufbau beschrieben.

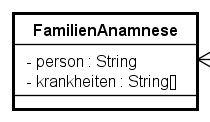
### Patient



Ein *Patient* ist die Repräsentation der Stammdaten eines Patienten. Diese Stammdaten umfassen neben Sozialversicherungsnummer, Name und Adresse auch die Sozialanamnese, Familienanamnese sowie die bisherige Krankengeschichte. Über eine Verknüpfung mit hinterlegten Nahrungsmitteln, können Abneigungen oder Medizinische Kriterien zu Nahrungsmitteln hinterlegt werden. Beispiel:

* Patient mag kein Käse
* Patient ist Allergisch gegen Nüsse

### FamilienAnamnese



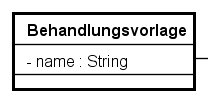
Eine *Familienanamnese* beinhaltet die Krankheiten für eine einzelne Person im Umfeld des Patienten. Die Krankheiten werden als Liste von beliebigen Texten abgelegt.

### Diätbehandlung



Eine *Diätbehandlung* gruppiert eine Reihe von Diätplänen mit dem Ziel einen über Diätparameter definierten Nutzen über einen definierten Zeitraum zu erreichen. Diese Behandlung kann mit der Umgangssprachlichen „Diät“ verglichen werden. Zu dieser Behandlung werden Protokolle unterschiedlicher Art erfasst. Ein Patient hat zu einem Zeitpunkt/-raum immer nur eine gültige Diätbehandlung. Es können nicht zwei oder mehr Diätbehandlungen zeitgleich verfügbar sein.

### Behandlungsvorlage



Oft benötigte Diätbehandlungen wie etwa eine „5 Wochen Diät“ können als *Vorlagen* hinterlegt werden. Diese Vorlagen können bei Bedarf als Basis für neue Diätbehandlungen dienen. Dieses Vorlagensystem ermöglicht es zusätzlich Vorlagen für einzelne Diätpläne zu hinterlegen.

### Behandlungsstatus



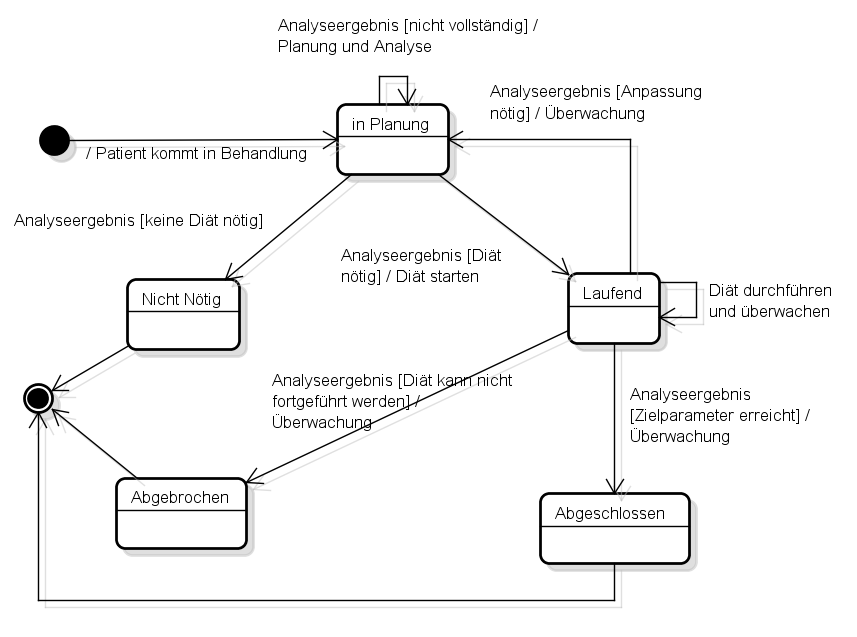
Der Behandlungsstatus gibt an in welcher Phase sich die Behandlung zurzeit befindet.

* In Planung  
  *Die Behandlung befindet sich im Planungsstadium. Es wurden noch keine Pläne zugeteilt. Der Patient ist evtl. mit dem ersten 24-Recall Protokoll beschäftigt.*
* Nicht Nötig

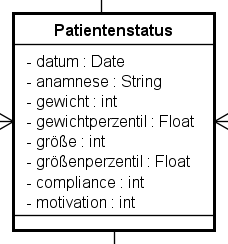
*Die Planungsphase hat ergeben dass der Patient keine Diät benötigt, die Diät wird nicht fortgeführt.*

* Laufend  
  *Die Diät läuft gerade. Der Patient ist mit der Umsetzung der Pläne und Erfassung von EPs beschäftigt.*
* Abgeschlossen  
  *Die Diät ist zeitlich abgeschlossen und wurde bis zum Ende durchgeführt.*
* Abgebrochen   
  *Die Diät wurde während der Durchführung abgebrochen.*

Das nachfolgende Diagramm beschreibt die Übergänge zwischen diesen einzelnen Status:

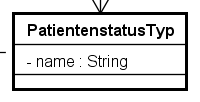


### Patientenstatus



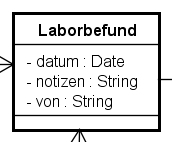
Im *Patientenstatus* wird der gesundheitliche und psychische Zustand des Patienten für einen Zeitpunkt als Fließtext erfasst. Zum Patientenstatus können Laborbefunde angehängt werden welche für diesen Status relevant sind. Der Patientenstatus wird über einen *PatientenstatusTyp* genauer typisiert.

### PatientenstatusTyp



Der *PatientenstatusTyp* definiert für einen Patientenstatus welche Art von Status erfasst wird. Dies kann neben einer *Zuweisungsdiagnose* auch eine *laufende Erfassung* sein.

### Laborbefund



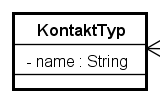
In einem Laborbefund werden die medizinischen Eigenschaften wie Blutwerte oder Körperfettanteil des Patienten erfasst. Der Laborbefund selbst beinhaltet dabei lediglich den *Kopf*  des Befundes. Die Werte zum Befund werden über eine Verknüpfung auf *Diätparameter* angelegt da die Struktur der Diätparameter die Erfassung von Werten für *Parameterdefinitionen* erlaubt.

### Kontaktjournal



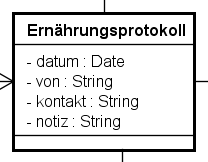
Im Kontaktjournal wird erfasst wann in welcher Form mit dem Patienten Kontakt aufgenommen wurde und auch warum. Die Kontakte werden zu einer Diätbehandlung erfasst.

### KontaktTyp



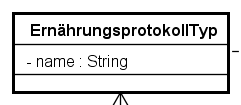
Der KontaktTyp legt für ein Kontaktjournal fest, um welche Art von Kontakt es sich gehandelt hat: Untersuchung, Telefonische Absprache, Diätplanüberarbeitung

### Ernährungsprotokoll



In einem *Ernährungsprotokoll* ist eine Repräsentation eines *Plans* welche dazu verwendet wird, verzehrte Mengen von Nahrungsmitteln zu erfassen. Über dieselbe Struktur eines normalen Plans kann über die Hierarchie von *Zeiträumen* und *Mahlzeiten* genau erfasst werden von welchem Nahrungsmittel/Rezept wie viel verzehrt wurde.

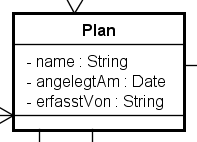
### ErnährungsprotokollTyp



Ein *ErnährungsprotokollTyp* definiert für ein Ernährungsprotokoll um welche Art von EP es sich handelt: Normales Ernährungsprotokoll, Wäge-Protokoll,…

Dieser Typ legt den Bindungsgrad fest, in welchem Detailgrad Protokolle erfasst werden müssen.

### Plan



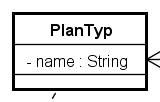
Ein *Plan* wird dazu verwendet um über eine Liste von Zeiträumen einen Diätplan, eine Ernährungsempfehlung oder ähnliches abzubilden. Über Diätparameter können Kriterien für diesen Plan festgelegt werden welche einzuhalten oder zu überwachen sind. Der zeitliche Gültigkeitsbereich des Plans wird über sogenannte *Zeiträume* festgelegt.

Pläne werden in Folge über diese Hierarchie befüllt:

* Pläne beinhalten Zeiträume
  + Zeiträume beinhalten Mahlzeiten
    - Mahlzeiten beinhalten Mahlzeitenzeilen
      * Mahlzeitenzeilen haben Nahrungsmittel/Rezepte zugeordnet

Um welche Art Plan es sich handelt, wird über einen Zugordneten Typ festgelegt.

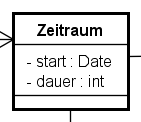
### PlanTyp



Der *PlanTyp* legt fest um welche Art von Plan es sich handelt. Je nach *PlanTyp* hat dies Auswirkung auf die Validierung und den Bindungsgrad des Planes:

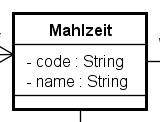
* Ernährungsempfehlung  
  *Der Patient hat sich lediglich an Diätparameter zu halten und bekommt keine genauen Nahrungsmittel vorgeschrieben*
* Diätplan  
  *Beim Diätplan kann der Patient sich zwischen den vorgegebenen Nahrungsmitteln und Rezepten aussuchen welche er zu sich nehmen möchte.*
* Speiseplan  
  *Der Speiseplan gibt dem Patienten exakt vor welche Nahrungsmittel er in welcher Menge verzehren muss. Der Patient hat sich daran zu halten.*

### Zeitraum



Ein *Zeitraum* bildet die erste Hierarchieebene innerhalb eines *Plans*. Ein Zeitraum definiert einen Startzeitpunkt und eine Dauer in Tagen für welche der Zeitraum gültig ist. Zeiträume dürfen sich nicht überschneiden (auch nicht über Pläne hinweg). Zu jedem Zeitpunkt darf nur maximal ein Zeitraum gültig sein. Für Zeiträume können, wie bei Plänen, optional Diätparameter festgelegt werden welche für diesen Zeitraum gültig sein müssen.

### Mahlzeit



Eine *Mahlzeit* ist eine für einen Zeitraum spezifische Ansammlung von Rezepten welche über *Mahlzeitenzeilen* hinzugefügt werden. Eine Mahlzeit hat neben einem Namen und Identifikationscode ebenfalls die Möglichkeit über Diätparameter den Nutzen der Mahlzeit genauer eingrenzen zu können.

### MahlzeitenZeile



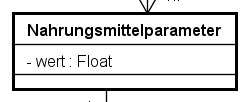
Eine *MahlzeitenZeile* ist ein einzelner Eintrag einer Mahlzeit in welcher für ein Rezept zusätzliche Angaben gemacht werden können. So kann neben der Verzehrmenge und einem Zusatztext auch festgelegt werden ob ein Rezept/Nahrungsmittel hinzugefügt oder entfernt werden soll. MahlzeitenZeilen können hierarchisch angeordnet sein was es erlaubt Varianten von Nahrungsmitteln zu erstellen:  
+ 1 Tasse Kaffee   
            - Zucker  
            + Süßstoff  
+ 1 Teller Gemischter Salat   
            - Rote Bete

### Rezept / Nahrungsmittel



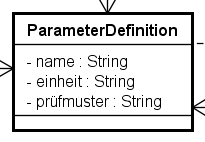
Ein *Rezept* beschreibt ein zu verzehrendes Produkt welches aus beliebig vielen weiteren Rezepten oder Nahrungsmitteln zubereitet und verzehrt werden kann. Rezepte und Nahrungsmittel werden größtenteils vom Bundeslebensmittelschlüssel vorgegeben und haben somit einen eindeutigen Code genannt BLS-Code zugeordnet. Der Schwierigkeitsgrad wird als 1-5 angegeben wobei 1 Leicht und 5 Schwierig ist. Der Nutzen beschreibt den Medizinischen Nutzen des Nahrungsmittels/Rezeptes als Fließtext während die Beschreibung eine inhaltliche Beschreibung des Rezeptes ist.

### Nahrungsmittelparameter



Ein *Nahrungsmittelparameter* beschreibt die Zuordnung einer Parameterdefinition inklusive Wert zu einem Rezept. Somit kann bspw. einem Nahrungsmittel Ei über einen Nahrungsmittelparameter ein Brennwert von 93 kcal zugewiesen werden.

### Parameterdefinition

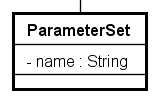


Eine Parameterdefinition beschreibt Nährwerte, Inhaltsstoffe,… deren Einheit und gültiger Wertebereich. Beispiele:

|  |  |  |
| --- | --- | --- |
| Name | Einheit | Gültige Werte |
| Brennwert | kcal | Zahlen |
| Vitamin B | g | Zahlen |
| Laktose |  | Frei, Haltig |

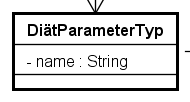
Die gültigen Werte werden über als Prüfmuster angegeben.

### Parameterset



Über ein *ParameterSet* können Parameterdefinitionen für die Wiederverwendung gruppiert werden. Während des Anlegens eines Rezeptes oder eines Diätparametersets können auf diese Parametersets zurückgegriffen werden um schnell alle nötigen Parameterdefinitionen hinzuzufügen.

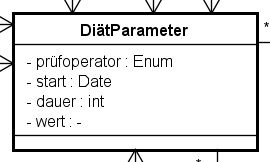
### DiätparameterTyp



Ein Diätparametertyp spezifiziert für einen Diätparameter welche Art von Diätparameter ist. Beispiele:

* Überwachungsparameter
* Zielparameter

### Diätparameter

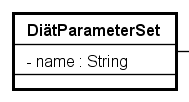


Ein *Diätparameter* definiert eine Prüfbedingung für eine Parameterdefinition. Ein Diätparameter kann somit festlegen welche Werte für welche Parameterdefinition in welchem Zeitraum einzuhalten sind. Beispiel:

* Von 01.01.2011 bis 05.01.2011 muss der Brennwert kleiner als 5000kcal sein.

Diese Parameter können in unterschiedlichen Bereichen des Systems verwendet werden um bspw. die Grenzwerte für Überwachungsparameter oder Zielparameter festzulegen. Die Datumsangabe solcher Parameter ist optional.

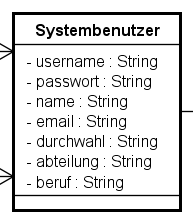
### DiätparameterSet



Ein Diätparameterset gruppiert eine Reihe von verwendbaren Diätparametern. So können Gruppen für Diätparameter angelegt werden die in Kombination oft benötigt werden. Die Angabe der Werte für Diätparameter in Diätparametersets ist optional. Beim Zuweisen eines *DiätParameterSets* werden alle Diätparameter dupliziert sodass Werte angepasst werden können.  
Beispiel:  
**Fettrelevante Parameter**

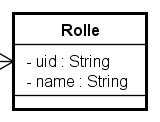
|  |  |  |
| --- | --- | --- |
| Parameterdefinition | Prüfoperator | Wert |
| Brennwert (kcal) | > |  |
| Brennwert (kcal) | < | 1500 |
| Fettgehalt (g) | < | 50 |

### Systembenutzer



Ein *Systembenutzer* hat die Möglichkeit sich bei EasyDiet mit seinen Anmeldedaten einzuloggen und Aktionen im System zu tätigen. Das Passwort wird im System verschlüsselt abgelegt.

### Rolle



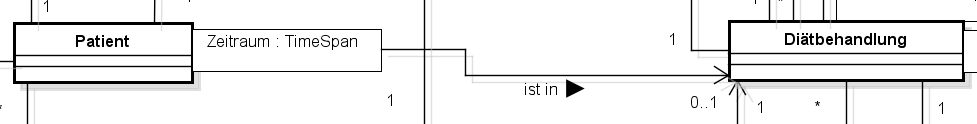
Eine Rolle räumt einem Systembenutzer Rechte innerhalb von EasyDiet ein. Diese Rollen werden auf Anwendungsfallebene abgebildet

Beispiele

|  |  |
| --- | --- |
| UID-Rolle | Beschreibung |
| CREATE\_PATIENT – Patientenstammdaten anlegen | Der Benutzer hat die Berechtigungen neue Patienten im System anzulegen |
| ADD\_RECIPE – Rezept anlegen | Der Benutzer hat die Berechtigungen neue Rezepte im System anzulegen. |

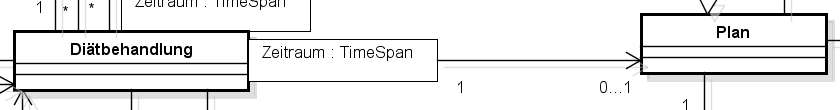
## Einschränkungen (Constraints)

### Patient 🡪 Diätbehandlung



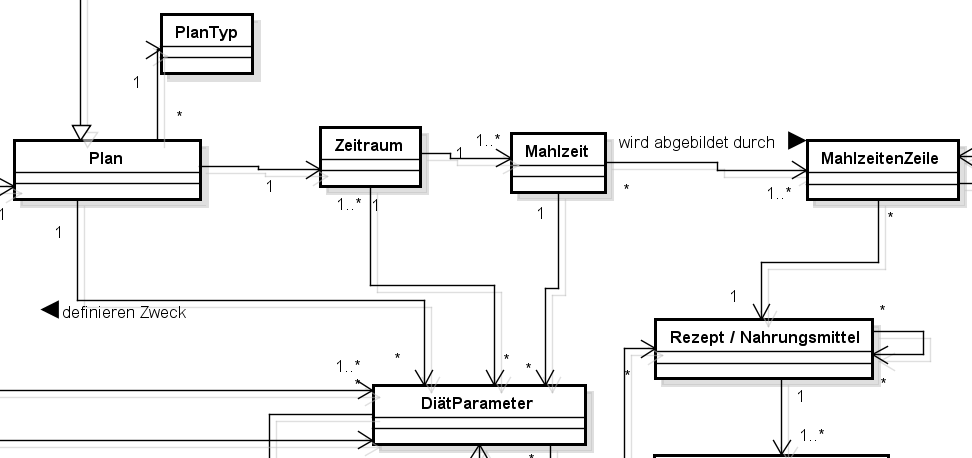
Der Patient hat innerhalb eines beliebigen Zeitraumes maximal eine Diätbehandlung zugeordnet. Diätbehandlungen dürfen Sich nicht zeitlich überschneiden.

### Diätbehandlung 🡪Plan



Innerhalb einer Diätbehandlung darf für jeden Zeitraum maximal ein Plan gültig sein.

### Mahlzeit 🡪 Diätparameter / Mahlzeitenzeile



Ein Spezialfall von Mahlzeiten tritt ein wenn die Mahlzeit innerhalb eines Plans vom Typ Ernährungsempfehlung angelegt wird. Ist dies der Fall, können zur Mahlzeit keine Mahlzeitenzeilen hinzugefügt werden. Stattdessen müssen nun zur Mahlzeit Diätparameter definiert werden. Im Spezialfall der Ernährungsempfehlung werden somit keine Nahrungsmittel vorgeschrieben, sondern es werden lediglich Kriterien für diese Mahlzeit festgelegt.

Ein weiterer Spezialfall hierbei ist: Wurden bereits im Zeitraum der Mahlzeit Diätparameter festgelegt ist die Angabe weiterer Diätparameter für die Mahlzeit keine Pflicht.

# Dynamisches Modell

Für alle Benutzungsfälle gilt:

* Der Benutzer welcher die Aktionen durchführt hat im System die nötigen Berechtigungen diese Aktionen auch durchzuführen.
* Sollte das System unerwartet Abstürzen, gilt dies als Abbruch des Benutzungsfalls.
* Der Benutzer kann jeden Benutzungsfall zu jedem beliebigem Schritt abbrechen.
* Beim Anlegen jeglicher Datensätze wird angenommen dass diese nach dem Speichern persistent und mit allen dafür nötigen Kriterien erfüllt im System abgelegt sind.

## Detaillierte Benutzungsfälle (UseCases)

### Systembenutzer anlegen

Vorbedingung

* Der Benutzer ist bereits am System angemeldet.

Main Success Szenario

1. Der Benutzer verwendet die verfügbare Suchmaske um die Liste der verfügbaren Benutzer zu filtern.
2. Der Benutzer sucht manuell in der Liste der verfügbaren Benutzern den gesuchten Eintrag
3. Der Benutzer wählt die Option Benutzerdetails anzeigen aus.
4. Das System zeigt die Detailansicht mit den Daten des Benutzers an.

### Patientenstamm anlegen

Vorbedingung

* Der Benutzer ist bereits am System angemeldet.

Main Success Szenario

1. Der Benutzer wählt die Option zur Erfassung von Patientendaten
2. Das System zeigt eine leere Patientendaten-Maske an.
3. Der Benutzer befüllt mithilfe des Patienten die allgemeinen Informationen des Patienten:
   * Sozialversicherungsnummer
   * Titel/Anrede, Vorname, Zuname, Geburtsdatum, Geschlecht,
   * Anschrift
4. Der Benutzer befüllt mithilfe des Patienten und die Sozialanamnese des Patienten
   * Familienstatus, Religion,…
5. Der Benutzer legt in der Kategorie Medizinische Infos alle früheren Krankheiten des Patienten und Ausschlusskriterien für Nahrungsmittel/Inhaltsstoffe an.
6. Der Benutzer legt in der Kategorie Familienanamnese eine neue Person an.
7. Das System stellt einen neuen Eintrag für die Familienanamnese einer Person dar.
8. Der Benutzer fügt der neuen Person eine Bezeichnung und vorangegangene oder aktuelle Krankheiten in dem neuen Eintrag von Schritt 4 hinzu.
9. Der Benutzer wiederholt Schritt 5-7 für alle relevanten Personen
10. Der Benutzer speichert die Patientendaten.
11. Das System bestätigt die Speicherung der Daten und wechselt auf die Patienten-Detailansicht.

Alternative Flows

3a. Der Benutzer befüllt mithilfe der E-Card des Patienten die allgemeinen Informationen des Patienten.

11a. Das System meldet einen Fehler bei der Speicherung der Daten.

1. Der Benutzer wiederholt Schritt 10 solange bis das Speichern erfolgreich war.
2. Der Benutzer benachrichtigt einen zuständigen Techniker über den Fehler.

11b. Das System meldet bei der Speicherung einen Fehler über falsch oder nicht ausgefüllte Daten.

1. Der Benutzer korrigiert seine Eingaben und wiederholt Schritt 10.

Nachbedingung

* Die Patientendaten sind persistent und korrekt im System hinterlegt.

### Diät anlegen

Vorbedingung

* Der Diätassistent ist bereits am System angemeldet
* Die Patientendaten sind bereits im System vorhanden.
* Das 24 Stunden Protokoll ist bereits im System vorhanden.
* Die Zuweisungsdiagnose ist bereits im System vorhanden.
* Der Diätassistent hat bereits den Anwendungsfall „Patientenstamm anzeigen“ durchgeführt.
* Der/Die Diätassistent/In wird in Folge nur noch DA genannt.

Main Success Szenario

1. Das System zeigt eine leere Maske zum Anlegen einer neuen Diät für den Patienten an.
2. Der DA legt die allgemeinen Daten der Diät fest:
   * Von-Bis Zeitraum

(Pflichtfeld; Bedingung: Von Datum vor Bis Datum)

* + Kurzbeschreibung

(Pflichtfeld)

* + Beschreibungstext

(Optional)

1. Der DA hängt an die Diät die zugehörige Zuweisungsdiagnose
2. Der DA hängt an die Diät das zugehörige 24-Stunden Protokoll
3. Der DA legt die Zielparameter der Diät fest.
4. Der DA speichert die Diät
5. Das System bestätigt die Speicherung der Daten und wechselt zur Diätdetailansicht.

Alternative Flows

\*a. Der DA bricht das Anlegen der Diät ab.

\*b. Der DA ist eine beliebige andere Person mit den nötigen Rechten.

7a. Das System meldet eine invalide Eingabe und markiert die zugehörigen Felder.

1. Der DA korrigiert die invaliden Eingaben. Weiter mit 6.

Nachbedingung

* Die Diät ist persistent zum Patienten zugeordnet im System abgelegt.
* Alle Daten der Pflichtfelder sind verfügbar.

Kontrakte

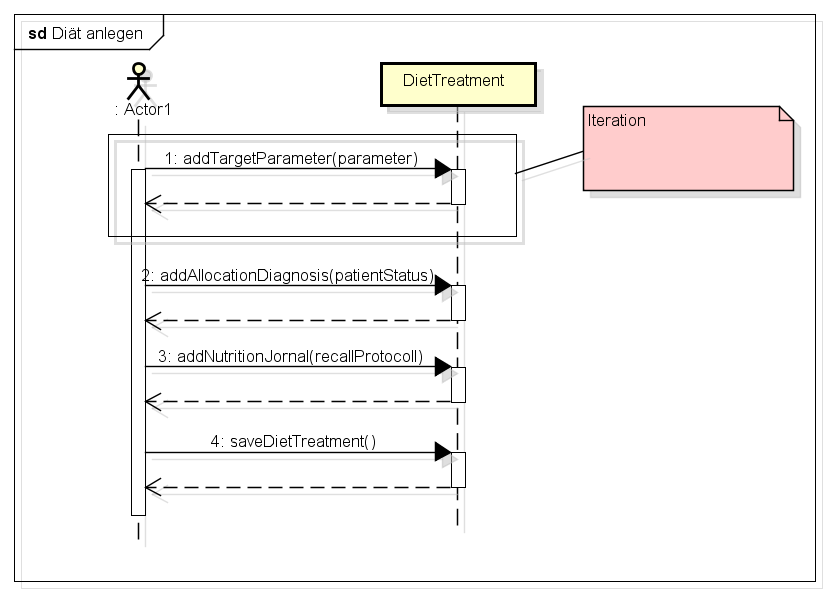
|  |  |
| --- | --- |
| Operation | addAllocationDiagnosis(in diet: DietTreatment, in patientStatus:PatientStatus) |
| Querverweise | Use Case: Diät anlegen |
| Vorbedingung | Das Objekt patientStatus ist wurde erzeugt und ist vom Typ Diagnosis |
| Nachbedingungen | Das Objekt patientStatus wurde der Diät zugewiesen. |

|  |  |
| --- | --- |
| Operation | addNutritionJornal(in diet: DietTreatment, in recallProtokoll: NutritionJornal) |
| Querverweise | Use Case Diät anlegen |
| Vorbedingung | Das Objekt recallProtokoll wurde erzeugt und ist vom Typ RecallProtokoll. |
| Nachbedingungen | Das Objekt recallProtokoll wurde der Diät zugewiesen. |

|  |  |
| --- | --- |
| Operation | addTargetParameter(in parameter: Parameter, in diet: DietTreatment) |
| Querverweise | Use Case Diät anlegen |
| Vorbedingung | Ein Objekt parameter wurde erzeugt |
| Nachbedingungen | Das Objekt parameter wurde der Diät zugewiesen |

|  |  |
| --- | --- |
| Operation | saveDietTreatment(in diet: DietTreatment) |
| Querverweise | Use Case Diät anlegen |
| Vorbedingung | Die persönlichen Daten wurden eingetragen. Die Methode addAllocationDiagnosis wurde ausgeführt. Die Methode addNutritionJornal wurde ausgeführt. Die Methode addTargetParameter wurde für allen gewünschten Zielparameter ausgeführt. |
| Nachbedingungen | Die persönlichen Daten wurden persistent im System gespeichert. Die Zuweisungsdiagnose wurde persistent im System gespeichert. Das 24-Stundenprotokoll wurde persistent im System gespeichert. Die Zielparameter wurden persistent im System gespeichert. |

Sequenzdiagramm



### Ernährungsprotokoll anlegen

Vorbedingung

* Der Anwender ist bereits am System angemeldet.
* Der Anwender hat die nötigen Rechte EPs zu erfassen.
* Der Anwender hat bereits den Anwendungsfall „Diät anzeigen“ durchgeführt.
* Die Protokolldaten sind verfügbar.
* Der durchführende Anwender wird in Folge nur noch User genannt.

Main Success Szenario

1. Das System zeigt eine leere Maske zur Erfassung eines EPs an.
2. Der User gibt die Daten für den EP Kopf ein.
   * Datum der Erfassung
   * Erfasst von
   * Protokolltyp
   * Parameterset
   * Kontakt der Erfassung
   * Notizen
3. Der User wählt die Option zum Hinzufügen eines neuen Zeitraumes
4. Das System zeigt den neuen Zeitraum an.
5. Der User wählt die Option zum Hinzufügen einer neuen Mahlzeit.
6. Das System Fragt nach dem Code und dem Namen der Mahlzeit.
7. Der User gibt Code und Name der Mahlzeit an und bestätigt.
8. Das System zeigt die Maske zum Hinzufügen von Rezepten und Nahrungsmitteln an.
9. Der User wählt die Option zum Hinzufügen eines neuen Rezeptes und Nahrungsmittels.
10. Das System zeigt eine Liste von Rezepten und Nahrungsmitteln an.
11. Der User wählt laut Protokoll ein Rezept oder Nahrungsmittel aus und bestätigt.
12. Der User trägt die laut Protokoll verzehrte Menge ein.
13. Der User wiederholt Schritt 10-12 bis Mahlzeit vollständig ist.
14. Der User wiederholt Schritt 5-13 für alle Mahlzeiten des Zeitraumes
15. Der User wiederholt Schritt 3-14 für alle Zeiträume des Protokolls.
16. Der User speichert die Daten ab.
17. Das System validiert die Eingaben auf Ihre Vollständigkeit und Korrektheit
18. Das System meldet das erfolgreiche Speichern der Daten.

Alternative Flows

\*a. Der User bricht das Anlegen des EPs ab.

5a. Der User wählt die Option zur Übernahme von Mahlzeiten des im Zeitraum gültigen Diätplans.

1. Das System sucht den Diätplan welcher am angegebenen Zeitraum gültig war und kopiert alle Mahlzeiten und Nahrungsmitteln in die Maske.
2. Der User fügt zusätzlich verzehrte Mahlzeiten zu den Mahlzeiten hinzu. Weiter mit 12.

17a. Das System meldet eine invalide Eingabe und markiert die zugehörigen Felder.

1. Der User korrigiert die invaliden Eingaben. Weiter mit 16.

Kontrakte

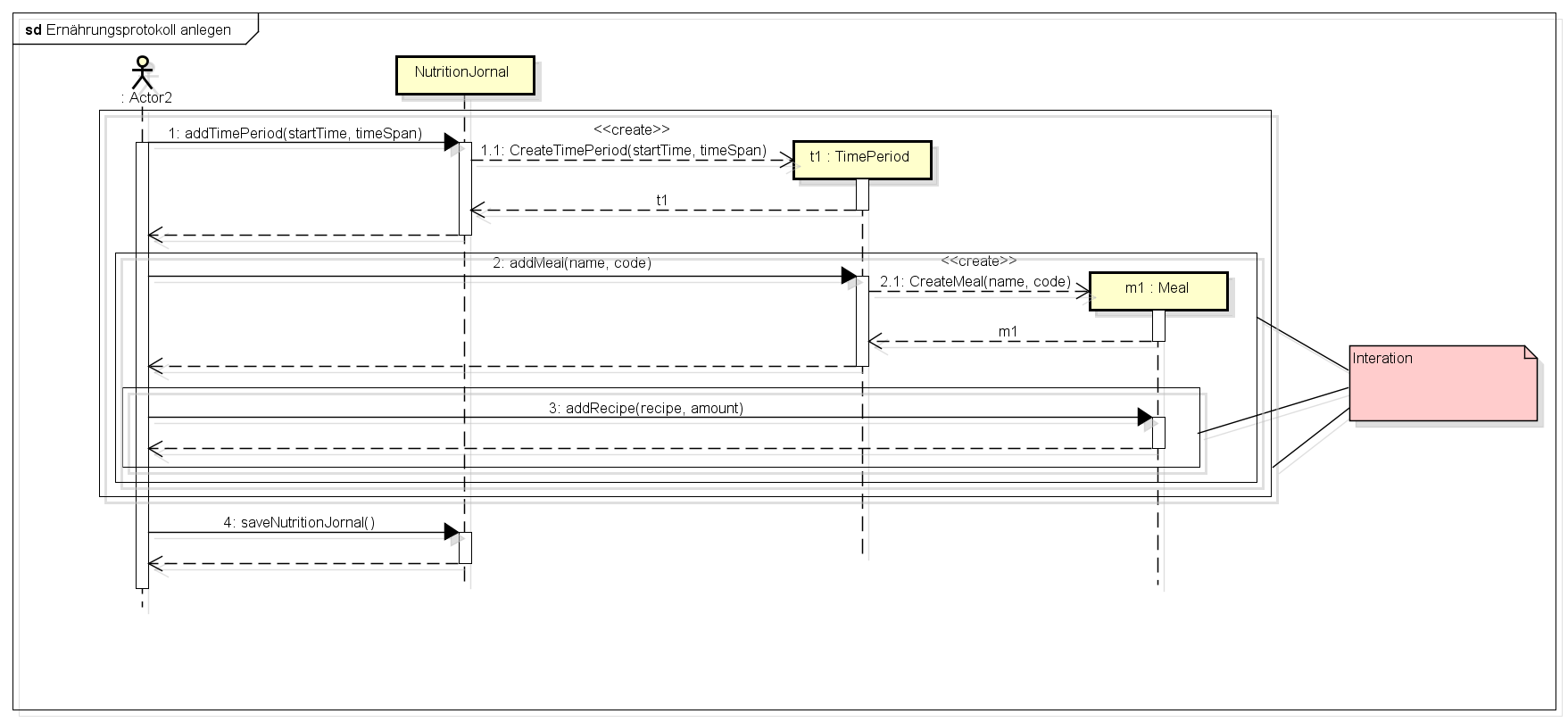
|  |  |
| --- | --- |
| Operation | addTimePeriod(in startDate: Datetime, in timeSpan: Integer, in nutritonJornal: NutritionJornal) |
| Querverweise | Use Case: Diätprotokoll anlegen |
| Vorbedingung | Ein NutritionJornal Objekt wurde erstellt. |
| Nachbedingungen | Ein Objekt TimePeriod wurde dem Ernährungsprotokoll zugewiesen. |

|  |  |
| --- | --- |
| Operation | addMeal(in name: String, in Code: String, in timePeriod: Timeperiod) |
| Querverweise | Use Case: Diätprotokoll anlegen |
| Vorbedingung | Ein TimePeriod Objekt wurde erstellt. |
| Nachbedingungen | Ein Objekt Meal wurde dem Diätprotokoll zugewiesen. |

|  |  |
| --- | --- |
| Operation | addRecipe(in recipe: Recipe, in amount: Integer, in meal: Meal) |
| Querverweise | Use Case: Diätprotokoll anlegen |
| Vorbedingung | Ein Mahlzeitobjekt wurde erstellt |
| Nachbedingungen | Ein Recipe Objekt wurde der Mahlzeit zugewiesen. Die Menge des verzehrten Rezeptes wurde dem Rezept zugewiesen |

|  |  |
| --- | --- |
| Operation | saveNutritionJornal(in nutritionJornal: NutritionJornal) |
| Querverweise | Use Case: Diätprotokoll anlegen |
| Vorbedingung | Die Methode addTimePeriod wurde für ausgeführt. Die Methode addMeal wurde ausgeführt. Die Methode addRecipe wurde ausgeführt. |
| Nachbedingungen | Alle TimePeriod Objekte wurden persistent gespeichert. Alle Meal Objekte wurden persistent gespeichert. Alle Recipe Objekte wurden persistent gespeichert. Alle Mengenangaben der verzehrten Rezepte wurden gespeichert. |

Sequenzdiagramm



### Rezepte anlegen

Vorbedingung

* Der Benutzer muss am System angemeldet sein.
* Der Benutzer muss die Berechtigungen haben Rezepte verwalten zu können.
* Es müssen bereits Nährstoffe im System vorhanden sein.

Main Success Szenario

1. Das System liefert eine leere Rezepteingabemaske
2. Der Benutzer gibt die allgemeinen Informationen zum Rezept ein:
   * Name (Pflichtfeld)
   * Schwierigkeitsgrad (Mit 1-5 Kochlöffeln)
   * Zubereitungszeit (Pflichtfeld)
   * Nutzungsbeschreibung (Pflichtfeld)

„Was hat dieses Rezept für eine Auswirkung“

* + Kurzbeschreibung (Optional)

1. Der Benutzer legt die Parameter des Rezeptes fest (10g Fett, 5kcal)
2. Der Benutzer wählt einen Bestandteil des Rezeptes aus einer Liste aus und fügt sie dem Rezept hinzu. (Brot, Nudeln, …)
3. Der Benutzer wiederholt Schritt 3-4 bis das Rezept vollständig ist.
4. Der Benutzer trägt die Zubereitungsanleitung ein.
5. Der Benutzer speichert das Rezept.
6. Das System bestätigt die Speicherung der Daten und wechselt auf die Rezept-Detailansicht.

Alternative Flows

\*a) Der Benutzer bricht den Vorgang ab.

6a. Der Benutzer gibt keine Zubereitungsanleitung ein da dieses nicht erforderlich ist.

8a. Das System meldet einen Fehler bei der Speicherung der Daten.

1. Der Benutzer wiederholt Schritt 8 solange bis das Speichern erfolgreich war.
2. Der Benutzer benachrichtigt einen zuständigen Techniker über den Fehler.

8b. Das System meldet bei der Speicherung einen Fehler über falsch oder nicht ausgefüllte Daten.

1. Der Benutzer korrigiert seine Eingaben und wiederholt Schritt 4.

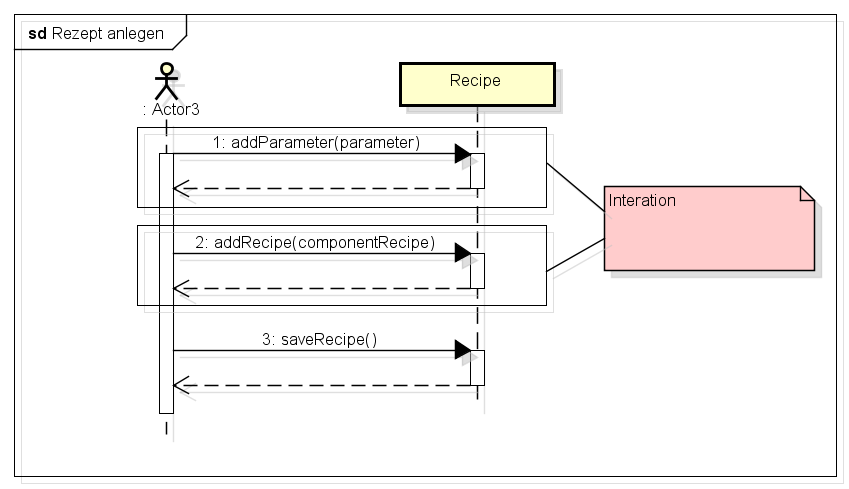
Kontrakte

|  |  |
| --- | --- |
| Operation | addParameter(in parameter: Parameter, in recipe: Recipe) |
| Querverweise | Use Case: Rezept anlegen |
| Vorbedingung | Ein Recipe Objekt muss angelegt sein. Eine Instanz parameter vom Objekt Parameter wurde angelegt |
| Nachbedingungen | Ein Objekt recipe wurde dem Rezept zugewiesen |

|  |  |
| --- | --- |
| Operation | addRecipe(in componentRecipe: Recipe, in recipe: Recipe |
| Querverweise | Use Case: Rezept anlegen |
| Vorbedingung | Ein Recipe Objekt muss angelegt sein Eine Instants componentRecipe eines Teilrezeptes wurde angelegt |
| Nachbedingungen | Ein Objekt componentRecipe wurde dem Rezept zugewiesen. |

|  |  |
| --- | --- |
| Operation | saveRecipe(in recipe: Recipe) |
| Querverweise | Use Case: Rezept anlegen |
| Vorbedingung | Die Methode addParameter wurde für ausgeführt. Die Methode addRecipe wurde für alle Teilrezepte ausgeführt. |
| Nachbedingungen | Alle allgemeinen Informationen des Rezeptes wurden persistent gespeichert. Alle Parameter Objekte wurden persistent gespeichert. Alle Recipe Objekte wurden persistent gespeichert. Die Zubereitungsanleitung wurde gespeichert. |

Sequenzdiagramm



### Patientenstatus anlegen

[TODO]

### Laborbefund anlegen

[TODO]

### Diätplan anlegen

Vorbedingung

* Der Benutzer ist bereits am System angemeldet.
* Es ist bereits eine Behandlung im System angelegt.

Main Success Szenario

1. System liefert eine leere Diätplan-Maske.
2. Der Benutzer legt einen neuen Zeitraum an.
3. Der Benutzer gibt das Von-Datum sowie das Bis-Datum oder die Länge des Zeitraums (1Tag, 1Woche,…) für welchen er die Mahlzeiten festlegen will ein.
4. Das System validiert die Datumseingabe darauf dass sich Zeiträume nicht überschneiden und meldet das Ergebnis dem Benutzer.
5. Der Benutzer fügt einen Zielparameter zum Zeitraum hinzu. („Was will ich in diesem Zeitraum erreichen“)
6. Der Benutzer wiederholt Schritt 5 solange bis alle gewünschten Zielparameter hinzugefügt sind.
7. Der Benutzer wählt die Option zum Hinzufügen einer neuen Mahlzeit zum Zeitraum.
8. Das System fragt nach dem Code oder Namen der Mahlzeit.
9. Benutzer gibt Code oder Name der Mahlzeit an und bestätigt.
10. Das System vervollständigt anhand des eingegebenen Codes oder Namens die Eingabe.
11. Das System zeigt die Maske zum Hinzufügen von Rezepten und Nahrungsmitteln. In dieser Maske befindet sich eine für den Patienten gefilterte Rezept/Nahrungsmittelliste.
12. Der Benutzer fügt Zielparameter zur Mahlzeit hinzu. („Was will mit dieser Mahlzeit erreichen“)
13. Der Benutzer wiederholt Schritt 12 solange bis alle gewünschten Zielparameter hinzugefügt sind.
14. Der Benutzer wählt nach seiner Erfahrung ein oder mehrere Rezepte/Nahrungsmittel aus und wählt die Option zum Hinzufügen der Auswahl zur Mahlzeit.
15. Das System überprüft anhand der gesetzten Zielparameter ob die aktuelle Nahrungsmittel- und Rezeptkonstellation im erlaubten Rahmen liegt und meldet das Ergebnis dem Benutzer.
16. Der Benutzer wiederholt Schritt 14-15 bis Mahlzeit vollständig ist.
17. Der Benutzer wiederholt Schritt 7-16 für alle Mahlzeiten des Tages
18. Der Benutzer wiederholt Schritt 2-17 für alle gewünschten Zeiträume des Diätplans.
19. Der Benutzer speichert den Diätplan.
20. Das System speichert die Daten und legt nicht vorhandene Mahlzeiten-Bezeichnungen mit einem neuen Code an.
21. Das System meldet ein erfolgreiches Speichern der Daten und wechselt zur Detailansicht des Diätplans.

Alternative Flows

\*a) Der Benutzer wählt die Option zum Speichern des Diätplans im Diätplanpool.

1. Das System fragt nach dem Namen und eine zugehörige Diät für den Plan im Diätpool
2. Der Benutzer gibt einen Namen und die zugehörige Diät für den Diätplan ein. (Gibt der Benutzer keine Diät an, wird der Diätplan einer „Standard-Diät“ im Pool zugewiesen)
3. Das System speichert eine Kopie des konfigurierten Diätplans ohne Von-Datums im
4. Diätplanpool und meldet das Ergebnis des Speichervorgangs.

5a. Der Benutzer fügt keine Zielparameter hinzu. Weiter mit 7.

10a. Der eingegebene Name oder Code wird nicht erkannt und nicht vervollständigt, die Eingabe bleibt bestehen.

12a. Der Benutzer fügt keine Zielparameter zur Mahlzeit hinzu (nicht erlaubt bei Ernährungsempfehlungen). Weiter mit 14.

14a. Der Benutzer fügt kein Rezept oder Nahrungsmittel hinzu (nur erlaubt bei Ernährungsempfehlungen) Weiter mit 17.

20a. Das System meldet einen Fehler bei der Speicherung der Daten.

1. Der Benutzer wiederholt Schritt 19 solange bis das Speichern erfolgreich war.
2. Der Benutzer benachrichtigt einen zuständigen Techniker über den Fehler.

20b. Das System meldet bei der Speicherung einen Fehler über falsch oder nicht ausgefüllte Daten.

1. Der Benutzer korrigiert seine Eingaben und wiederholt Schritt 19.

Nachbedingung

* Der Diätplan ist persistent im System gespeichert.
* Der Diätplan verletzt keine Regeln der festgelegten Zielparameter
* Der Patient ist auf die Wünsche und Bedürfnisse des Patienten abgestimmt.
* Der Diätplan erfüllt die Erfordernisse der Diät.

Kontrakte

|  |  |
| --- | --- |
| Operation | addTimePeriod(in startDate: Datetime, in timeSpan: Integer, in dietPlan: DietPlan) |
| Querverweise | Use Case: Diätplan anlegen |
| Vorbedingung | Ein Diätplanobjekt wurde erstellt. |
| Nachbedingungen | Ein TimePeriod Objekt wurde dem Diätplan zugewiesen |

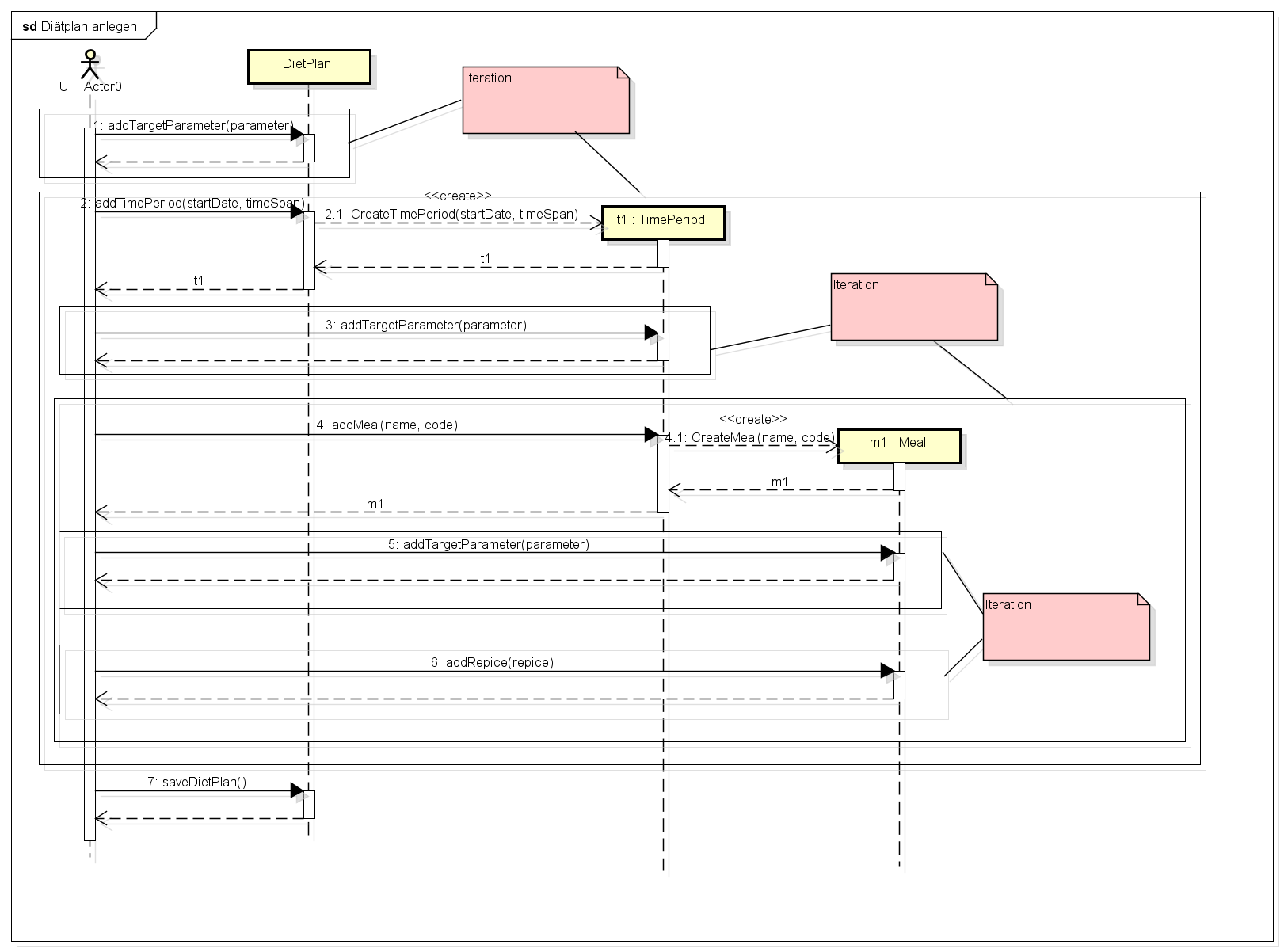
|  |  |
| --- | --- |
| Operation | addTargetParameter(in parameter: Parameter, in rage: Range) |
| Querverweise | Use Case: Diätplan anlegen |
| Vorbedingung | Ein Diätplanobjekt wurde erstellt |
| Nachbedingungen | Ein TargetParameter wurde dem Diätplan, dem Zeitraum oder der Mahlzeit zugewiesen |

|  |  |
| --- | --- |
| Operation | addMeal(in name: String, in Code: String, in timePeriod: Timeperiod) |
| Querverweise | Use Case: Diätplan anlegen |
| Vorbedingung | Ein TimePeriod Objekt wurde erstellt Ein Diätplanobjekt wurde erstellt |
| Nachbedingungen | Ein Meal Objekt wurde einem Zeitraum zugewiesen. |

|  |  |
| --- | --- |
| Operation | addRecipe(in recipe: Recipe, in meal: Meal) |
| Querverweise | Use Case: Diätplan anlegen |
| Vorbedingung | Ein Mahlzeitobjekt wurde erstellt |
| Nachbedingungen | Ein Recipe Objekt wurde der Mahlzeit zugewiesen. |

|  |  |
| --- | --- |
| Operation | saveDietaryPlan(in dietPlan: DietPlan) |
| Querverweise | Use Case: Diätplan anlegen |
| Vorbedingung | Die Methode addTimePeriod wurde ausgeführt. Die Methode addTargetParameter wurde ausgeführt, wenn Zielparameter gewünscht sind.  Die Methode addMeal wurde ausgeführt. Die Methode addRecipe wurde ausgeführt |
| Nachbedingungen | Alle TimePeriod Objekte wurden persistent gespeichert. Alle TargetParameter Objekte wurden persistent gespeichert. Alle Meal Objekte wurden persistent gespeichert. Alle Recipe Objekte wurden persistent gespeichert. |

Sequenzdiagramm



### Parameterset anlegen

Vorbedingung

* Der Benutzer ist bereits am System angemeldet.

Main Success Szenario

1. Das System liefert eine leere Maske zum Anlegen eines neuen Parametersets.
2. Der Benutzer gibt einen Namen für das Parameterset ein.
3. Der Benutzer fügt einen Parameter zum Set hinzu.
4. Das System zeigt eine neue Zeile für die Konfiguration des Parameters an
5. Der Benutzer legt die Kriterien für den Parameter fest.
6. Das Systemprüft ob sich keine Kriterien der vorhandenen Parameter widersprechen.
7. Der Benutzer wiederholt die Schritte 3-6 bis alle gewünschten Parameter angelegt sind.
8. Der Benutzer speichert das Parameterset
9. Das System meldet das Speichern des Parametersets und wechselt zur Detailansicht des Parametersets.

Alternative Flows

9a. Das System meldet einen Fehler bei der Speicherung der Daten.

1. Der Benutzer wiederholt Schritt 8 solange bis das Speichern erfolgreich war.
2. Der Benutzer benachrichtigt einen zuständigen Techniker über den Fehler.

9b. Das System meldet bei der Speicherung einen Fehler über falsch oder nicht ausgefüllte Daten.

1. Der Benutzer korrigiert seine Eingaben und wiederholt Schritt 8.

### Diät im Diätpool anlegen

Vorbedingung

* Der Benutzer ist bereits am System angemeldet.

Main Success Szenario

1. Das System zeigt eine leere Maske zum Anlegen einer neuen Diät für den Diätpool an.
2. Der Benutzer legt die allgemeinen Daten der Diät fest:
   * Länge

(Pflichtfeld; 1Tag, 1Woche…)

* + Kurzbeschreibung

(Pflichtfeld)

* + Beschreibungstext

(Optional)

1. Der Benutzer legt die Zielparameter der Diät fest.
2. Der Benutzer speichert die Diät.
3. Das System bestätigt die Speicherung der Daten und wechselt zur Diätdetailansicht.

Alternative Flows

\*a. Der Benutzer bricht das Anlegen der Diät ab.

3a. Der Benutzer legt keine Zielparameter fest.

5a. Das System meldet eine invalide Eingabe und markiert die zugehörigen Felder.

1. Der Benutzer korrigiert die invaliden Eingaben. Weiter mit 4.

Nachbedingung

* Die Diät ist persistent im Diätpool des Systems abgelegt.
* Alle Daten der Pflichtfelder sind verfügbar.

### Diätplan im Diätpool anlegen

Vorbedingung

* Der Benutzer ist bereits am System angemeldet.
* Der Benutzer wählt eine bestehende Diät im Pool oder die allgemeine Diät.

Main Success Szenario

1. System liefert eine leere Diätplan-Maske.
2. Der Benutzer gibt den Namen für den Diätplan ein.
3. Der Benutzer legt einen neuen Zeitraum an.
4. Der Benutzer gibt die Länge des Zeitraums (1Tag, 1Woche,…) für welchen er die Mahlzeiten festlegen will ein.
5. Der Benutzer fügt einen Zielparameter zum Zeitraum hinzu. („Was will ich in diesem Zeitraum erreichen“)
6. Der Benutzer wiederholt Schritt 5 solange bis alle gewünschten Zielparameter hinzugefügt sind.
7. Der Benutzer wählt die Option zum Hinzufügen einer neuen Mahlzeit zum Zeitraum.
8. Das System fragt nach dem Code oder Namen der Mahlzeit.
9. Benutzer gibt Code oder Name der Mahlzeit an und bestätigt.
10. Das System vervollständigt anhand des eingegebenen Codes oder Namens die Eingabe.
11. Das System zeigt die Maske zum Hinzufügen von Rezepten und Nahrungsmitteln. In dieser Maske befindet sich eine für den Patienten gefilterte Rezept/Nahrungsmittelliste.
12. Der Benutzer fügt einen Zielparameter zur Mahlzeit hinzu. („Was will mit dieser Mahlzeit erreichen“)
13. Der Benutzer wiederholt Schritt 12 solange bis alle gewünschten Zielparameter hinzugefügt sind.
14. Der Benutzer wählt nach seiner Erfahrung ein oder mehrere Rezepte/Nahrungsmittel aus und wählt die Option zum Hinzufügen der Auswahl zur Mahlzeit.
15. Das System überprüft anhand der gesetzten Zielparameter ob die aktuelle Nahrungsmittel- und Rezeptkonstellation im erlaubten Rahmen liegt und meldet das Ergebnis dem Benutzer.
16. Der Benutzer wiederholt Schritt 14-15 bis Mahlzeit vollständig ist.
17. Der Benutzer wiederholt Schritt 7-16 für alle Mahlzeiten des Tages
18. Der Benutzer wiederholt Schritt 3-17 für alle gewünschten Zeiträume des Diätplans.
19. Der Benutzer speichert den Diätplan.
20. Das System speichert die Daten und legt nicht vorhandene Mahlzeiten-Bezeichnungen mit einem neuen Code an.
21. Das System meldet ein erfolgreiches Speichern der Daten und wechselt zur Detailansicht des Diätplans.

Alternative Flows

5a. Der Benutzer fügt keine Zielparameter hinzu. Weiter mit 7.

10a. Der eingegebene Name oder Code wird nicht erkannt und nicht vervollständigt, die Eingabe bleibt bestehen

12a. Der Benutzer fügt keine Zielparameter zur Mahlzeit hinzu (nicht erlaubt bei Ernährungsempfehlungen). Weiter mit 14.

14a. Der Benutzer fügt kein Rezept oder Nahrungsmittel hinzu (nur erlaubt bei Ernährungsempfehlungen) Weiter mit 17.

20a. Das System meldet einen Fehler bei der Speicherung der Daten.

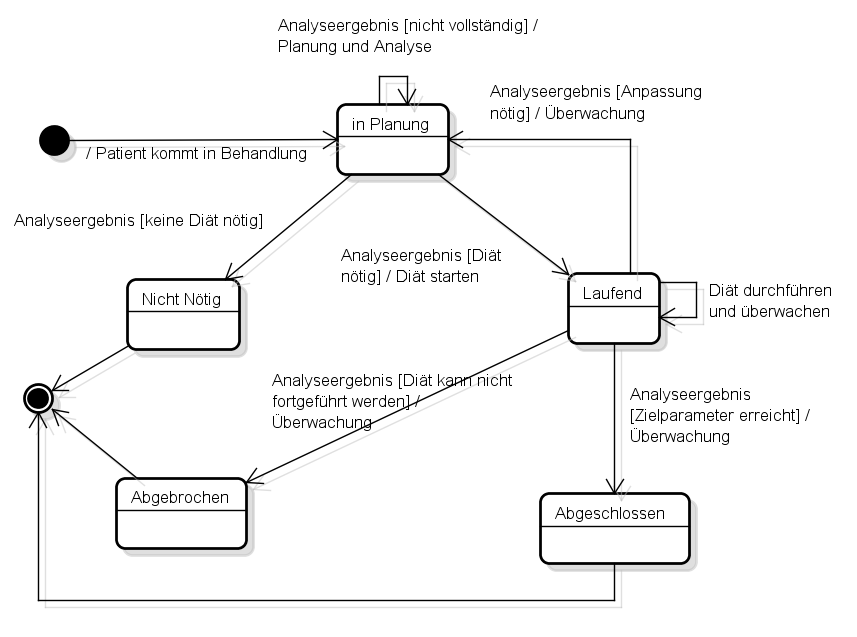
1. Der Benutzer wiederholt Schritt 19 solange bis das Speichern erfolgreich war.
2. Der Benutzer benachrichtigt einen zuständigen Techniker über den Fehler.

20b. Das System meldet bei der Speicherung einen Fehler über falsch oder nicht ausgefüllte Daten.

1. Der Benutzer korrigiert seine Eingaben und wiederholt Schritt 19.

## Objekt Lifecycles

### Lebenszyklus einer Diätbehandlung



# Nichtfunktionale Anforderungen

[nach den FURPS+ Kategorien, siehe Vorlesungsfolien Kapitel 2: Anforderungestypen. Nur das auswählen, was wir für den Roomanizer wirklich als Anforderung wissen]

## Regeln

Patienten werden nur behandelt wenn ihre Identität festgestellt wurde oder ein akuter Notfall ein umgehen dieser Regel erfordert.

Nur ein ausgebildeter Diätassistent darf Diätpläne verwalten, da es sonst zu rechtlichen Problemen für das Krankenhaus kommen kann.

## Usability

Das Programm wird eine intuitive Oberfläche besitzen, das heißt man muss sich nicht zuerst mit der Dokumentation befassen sondern kann gleich mit der Arbeit beginnen.

Das Programm unterstützt den User wo immer es geht mit Hilfe von Tooltips, Erklärungen, Validierungsfunktionen und Hinweisen.

Falsche oder nicht eingegeben Daten in einem Formular werden, sofern überprüfbar, auch als Falsch markiert und dem Nutzer wird eine Meldung dazu ausgegeben.

## Zuverlässigkeit

Benutzereingaben werden vom Benutzer manuell gespeichert.

Stürzt das Programm ab, kann mit den zuletzt gespeicherten Daten weitergearbeitet werden.

Grundsätzlich sollte das System stabil und ohne Abstürze funktionieren.

## Performanz

Das Programm soll dem/der DA die Möglichkeit bieten Diätpläne schneller zu erstellen.

Es werden dadurch mehr Patienten an einem Tag behandelt werden können.

## Unterstützbarkeit

EasyDiet ist für das AKH Wien ausgelegt, das heißt es wird nur in Deutsch angeboten.

Die Einträge der Tabellen in der Datenbank werden vom Benutzer mit Hilfe des Programms selbst administriert.

## Online Benutzerdokumentation und Help System

Die Benutzerdokumentation wird in englischer Sprache in folgenden Formaten verfügbar angeboten:

* Online Benutzerdokumentation
* PDF Dokument
* CHM Hilfe-Datei

## Zugekaufte Komponenten

Für die Entwicklung ist der BLS des Bundeministeriums für Ernährung, Landwirtschaft und Verbraucherschutz gratis.

Für die Verwendung im AKH muss dieser zugekauft werden. Hierfür bietet das Bundesministerium verschiedene Lizenzmodelle an. Nachzusehen sind diese auf der Homepage des Bundesministeriums (http://www.bls.nvs2.de/index.php?id=134).

## Schnittstellen

### Benutzerschnittstellen

[TODO:Desktop interface und Webinterface]

### Software Schnittstellen

Die Software bietet Schnittstellen zur Krankenhaussoftware. Diese ermöglicht den Austausch von Patientendaten, damit nicht jeder Patient neu im EasyDiet-System angelegt werden muss.

### Kommunikationsschnittstellen

[TODO: Was ist hier gemeint? Handschriftliche Formulare, Drucker, Scanner, E-Card reader,… (hardware)]

## Zusätzliche Lizenzierungen

Für die Realisierung dieses Projektes sind keine weiteren Lizensierungen, als die bisher genannten nötig.

## Copyright und andere rechtliche Anforderungen

Das Copyright der Software teilen sich die Entwickler und die Stadt Wien die als Systeminhaber das Projekt auch finanziert.

Die Software bleibt geistiges Eigentum der Entwickler wobei der Stadt Wien jegliche Rechte zum Vertrieb und Einsatz der Software zugestanden werden.

Das Krankenhaus muss selbst sicherstellen, dass nur berechtigte Personen die Zugangsdaten zum System erhalten um somit den gesetzlichen Bestimmungen Folge zu leisten.

Die Entwickler distanzieren sich von jeglicher Verantwortung die im Zusammenhang mit der Verwendung des Systems.

## Anzuwendende Standards

Es wurden keine Standards festgelegt.

# Iterationenplan (Timeboxes)

## Überblick

|  |  |  |
| --- | --- | --- |
| Timebox1 | Timebox2 | Timebox3 |
| 1. Patientenstamm anlegen, anzeigen, bearbeiten, löschen *Komplett* 2. Diät anlegen, anzeigen, bearbeiten löschen  *Komplett* 3. Diätpläne anlegen, anzeigen, bearbeiten löschen  *Komplett* | 1. Ernährungsprotokolle anlegen, anzeigen bearbeiten, löschen 2. Patientenstatus anlegen, anzeigen bearbeiten, löschen 3. Laborbefunde anlegen, anzeigen bearbeiten, löschen 4. Rezepte anlegen, anzeigen bearbeiten, löschen 5. Parameterset anlegen, anzeigen bearbeiten, löschen 6. Diät im Diätpool anlegen, anzeigen bearbeiten, löschen 7. Diätplan im Diätpool anlegen, anzeigen bearbeiten, löschen | 1. Systembenutzer anlegen, anzeigen, bearbeiten, löschen 2. Rollen eines Systembenutzers verwalten 3. Anmelden am System |

*Wenn nicht anders angegeben werden die angegebenen Benutzungsfälle komplett mit MainSuccess-Szenario und allen Extensions realisiert.*

## Timebox 1

*30.03.2011 – 27.04.2011*

In dieser Timebox wird die gesamte Basisstruktur für die Anwendung geschaffen. Das Datenbankdesign wird umgesetzt und die nötige Infrastruktur geschaffen um mit der Datenbank kommunizieren zu können.

In dieser Periode wird zudem der Fokus auf die zentralen Benutzungsfälle der Anwendung gelegt.

### Benutzungsfall/fälle (UseCase(s))

1. Patientenstamm anlegen, anzeigen, bearbeiten, löschen
2. Diät anlegen, anzeigen, bearbeiten löschen
3. Diätpläne anlegen, anzeigen, bearbeiten löschen

### Architektur

* Datenbankanbindung
* Domänenmodell
* GUI

### Deliverables

* Datenbankmodell
* Framework um die Daten persistent in der Datenbank ablegen zu können.
* Testergebnisse der Framework-Tests
* Datenklassen für das Arbeiten mit den Daten im Code
* Architekturspezifikation
* Klassendiagramm des Systems
* Grafische Benutzeroberfläche für die Benutzungsfälle

### Abhängigkeiten

* Signiertes Pflichtenheft
* Freigabe des Starts von den verantwortlichen Projektbetreuern

## Timebox 2

*27.04.2011 – 23.05.2011*

In dieser Timebox werden die noch offenen Benutzungsfälle realisiert. Ziel dieser Timebox ist die Realisierung möglichst vieler geplanter Programmfunktionen.

### Benutzungsfall/fälle (UseCase(s))

1. Ernährungsprotokolle anlegen, anzeigen bearbeiten, löschen
2. Patientenstatus anlegen, anzeigen bearbeiten, löschen
3. Laborbefunde anlegen, anzeigen bearbeiten, löschen
4. Rezepte anlegen, anzeigen bearbeiten, löschen
5. Parameterset anlegen, anzeigen bearbeiten, löschen
6. Diät im Diätpool anlegen, anzeigen bearbeiten, löschen
7. Diätplan im Diätpool anlegen, anzeigen bearbeiten, löschen

### Architektur

* Ausdrucke
* Datenbankanbindung
* Domänenmodell
* GUI

### Deliverables

* Desktop Interface für realisierte Benutzungsfälle
* Aktualisiertes Klassendiagramm

### Abhängigkeiten

* Datenbankmodell [Timebox 1]
* Framework um die Daten persistent in der Datenbank ablegen zu können.[Timebox 1]
* Datenklassen für das Arbeiten mit den Daten im Code [Timebox 1]
* Architekturspezifikation [Timebox 1]

## Timebox 3

*23.05.2011 – 10.06.2011*

In der Timebox 3 werden die bereits realisierten Anwendungsfälle von einer Desktopanwendung in eine Webanwendung übertragen. Weiterführend wird ein Authentifizierungssystem implementiert und die Dokumentationen geschrieben.

### Benutzungsfall/fälle (UseCase(s))

1. Systembenutzer anlegen, anzeigen, bearbeiten, löschen
2. Rollen eines Systembenutzers verwalten
3. Anmelden am System

### Architektur

* Reports, Ausdrucke, Networking, Security, HW-Plattform
* Datenbankanbindung
* Datenbankanbindung
* Domänenmodell

### Deliverables

* Webinterface für alle bereits realisierten Benutzungsfälle
* System für Feherbehandlung

### Abhängigkeiten

* Realisierte Benutzungsfälle [Timebox 1-2]
* Benutzerhandbuch
* Technische Dokumentation

# Glossar

[wichtige Begriffe des Domänenbereichs in kurzer Textform  definiert]