

Projektauftrag

QSE SEPM02 SS09



taco.

Eine Rezepte-Datenbank

Projektmitglieder

Rolle	Verantwortlicher	Stellvertreter
Teamkoordinator	Thomas Perl (0725603)	Stefan Kögl
Techn. Architekt	Stefan Kögl (0725134)	Thomas Perl
Dokumentenbeauftragter	Marco Augustin (0725924)	Alexander Duml
Testbeauftragter	Alexander Duml (0726471)	Martin Wieser
User Interface Designer	Christoph Zoller (0726845)	Marco Augustin
Datenbankbeauftragter	Martin Wieser (0727131)	Christoph Zoller

Ausgangssituation

Kochrezepte sind heutzutage nicht nur in traditionellen Kochbüchern sondern auch auf diversen Webseiten zu finden. Allerdings ergibt die Suche nach **Rezepten mit bestimmten Zutaten** meist nicht das gewünschte Ergebnis. So wird zwar eine grosse Anzahl an Rezepten gefunden, welche diese Zutaten enthalten, oft werden jedoch neben den eingegebenen Zutaten auch zusätzliche benötigt, die man möglicherweise nicht zu Hause hat und noch besorgt werden müssen.

Bei herkömmlichen Suchen wird den Zutaten keine Semantik verliehen. So findet man bei der Suche nach "Paradeiser" keine Rezepte mit Tomaten oder keine Hühnergerichte bei der Suche nach "Fleisch".

Auch ist es meistens nicht möglich eine eingegebene Zutat komplett auszuschließen oder eine Zutat durch eine andere zu ersetzen, wie z.B. Fleisch durch Tofu. Die Auswahl an Speisen für Vegetarier ist deswegen meistens sehr gering.

Projektbeschreibung

Es soll eine Software erstellt werden, die Kochrezepte verwaltet und dem Benutzer, nach Eingabe aktuell vorhandener Zutaten, Rezept-Vorschläge anbietet.

Bei diesen Rezept-Vorschlägen werden unterschiedliche Daten berücksichtigt.

- der virtuelle Kühlschrank, der vom Benutzer verwaltet werden kann
- neu eingegebene Zutaten
- gespeicherte Ähnlichkeiten von Zutaten

Anhand dieser Daten werden dem Benutzer Rezepte vorgeschlagen, die zuvor entweder manuell eingetragen oder aus dem Web importiert wurden. Beim Web-Import kann der Benutzer eine Website in der Applikation auswählen, von der er dann alle Rezepte in die Datenbank einbindet. Der Web-Import ist dabei so gestaltet, dass für neue Webseiten Plugins zur Verfügung gestellt werden.

Die angezeigten Vorschläge kann der Benutzer danach weiter einschränken, indem er beispielsweise alle Rezepte ausschließt, die eine bestimmte Zutat enthalten oder eine bestimmte Anzahl an Kalorien überschreiten.

Aus den angezeigten Rezepten kann der Benutzer eines auswählen und anzeigen lassen - wenn vorhanden, mit den Bildern vom Gericht. Ebenso soll das Programm eine Export Funktion haben, mit dieser der Anwender das ausgewählte Rezept in druckbarer Form erhält (PDF,XML).

Um eine hohe Vielfalt der gefundenen Rezepte zu garantieren, können vom Benutzer "ähnliche Zutaten" definiert werden. Beispielsweise kann ein Vegetarier "Fleisch" mit "Tofu" verbinden und erhält dann bei seiner Suche auch Fleisch-Gerichte angezeigt.

Des Weiteren kann der Benutzer neue Rezepte manuell eingeben und vorhandene Rezepte verändern.

Zusätzlich zur Suche nach Zutaten kann der Benutzer auch nach Rezept-Themen (asiatisch, indisch,...) suchen um zum gewünschte Ergebnis zu gelangen.

Die Zutatenmenge soll nach Eingabe der Personenanzahl umgerechnet werden können.

Nachdem sich der Benutzer eine Meinung vom Rezept gebildet hat, kann er diese mit Hilfe einer Bewertungsfunktion dem Programm mitteilen. Auch soll es dem Benutzer möglich sein statistische Auswertungen über sein Kochverhalten anzeigen zu lassen, wie z.B. die am öftesten gekochten Rezepte.

Es soll möglich sein, die Zutaten des ausgewählten Rezeptes mit dem Bestand des virtuellen Kühlschranks zu vergleichen, um anhand dieses Vergleichs eine Einkaufsliste zu erstellen die auch in PDF oder XML exportiert werden kann.

Der oben schon erwähnte virtuelle Kühlschrank soll auch mit einem Barcode- Scanner-Interface- bzw. RFID-Interface ausgestattet werden um ihn mit dem wirklichen Kühlschrank abzugleichen.

Projektabgrenzung

Folgende Features werden in unserem Projekt nicht gelöst und berücksichtigt. Diese Features können jedoch in einem möglichen Nachfolgeprojekt implementiert werden. Für dieses Projekt werden diese Features nicht implementiert.

- Gespeicherte Rezepte können nicht im Internet veröffentlicht werden.
- Es gibt keine Möglichkeit, einen Ernährungsplan zu erstellen oder eine Planung für eine gesunde Ernährung durchzuführen.
- Es erfolgt keinerlei Bewertung der eingegebenen Zutaten (empfohlener Tagesbedarf, etc...). Als optionales Feature kann die Kalorien-Anzahl für Rezepte berechnet werden.
- Das User Interface ist nur in deutscher Sprache verfügbar. Es ist keine Internationalisierung vorgesehen.
- Es ist nicht möglich den Inhalt des virtuellen Kühlschranks automatisch mit dem Inhalt des tatsächlichen Kühlschranks zu synchronisieren. Die für eine unterstützte Aktualisierung durch einen Barcode-Scanner nötige Schnittstelle wird als optionales Feature bereitgestellt.
- Die fehlenden Zutaten können nur als Einkaufsliste ausgedruckt werden. Das Programm kann keine Einkäufe im Internet tätigen und bietet außer dem Export keine weitere Schnittstelle, um fehlende Zutaten zu erwerben.
- Es ist nicht möglich, statistische Auswertungen aus dem Internet zu laden. Alle statistischen Auswertungen finden anhand der lokal durchgeführten Aktionen (manuelle Bewertung und automatische Protokollierung) statt.
- Die direkte Suche nach Rezepten im Internet ist nicht möglich. Es wird lediglich der Bulk-Import von Rezepten aus Webseiten in die lokale Datenbank unterstützt. Alle Abfragen und Editiervorgänge werden lokal durchgeführt.

Zielgruppen

Grundsätzlich richtet sich die Software an Personen, die gerne und oft eine Vielzahl von verschiedenen Menüs kochen und ausprobieren. Dieser Zielgruppe steht dabei eine übersichtliche Applikation mit einigen Features, die es in solch kompakter Form in sonstigen Medien nicht gibt, zur Verfügung. Zusätzlich kann diese Zielgruppe auch bereits vorhandene Rezepte, die vielleicht in einem unübersichtlichen Stapel über die Jahre gesammelt wurden, in das Programm eintragen. Die Folgerung wäre ein strukturierter Überblick über alle Rezepte.

Im Speziellen sollte jedoch auf Personen Rücksicht genommen werden, die aus gewissen Gründen (Unverträglichkeit, Diabetiker, Vegetarier, Veganer, ...) besonderen Wert auf die Auswahl ihrer Speisen legen müssen. Die Software verknüpft hierbei 'verträgliche' Zutaten wie z.B. Tofu mit 'nicht verträglichen' Zutaten wie z.B. Fleisch und berücksichtigt dies in der Suche um eine größere Auswahl an Speisen zu finden.

Im Weiteren richtet sich die Applikation an Gruppen, die für eine Mahlzeit so wenig wie möglich zusätzlich einkaufen wollen. Hierbei kann der Anwender seine schon im Kühlschrank vorhandenen Zutaten in die Software eingeben und sich auf Grund dieser einen Rezeptvorschlag erstellen lassen. Interessant hierbei wäre auch eine Verknüpfung des

Programmes mit Kühlschränken, die über einen Barcode Scanner verfügen und somit der virtuelle Kühlschrank des Programms immer über den Essensbestand des eigentlichen Kühlschranks Bescheid weiß. Im weiteren Sinne sind also auch Personen die über einen solchen Kühlschrank verfügen eine unserer Zielgruppen.

Feature-Liste

Die Feature-Liste beinhaltet Anforderungen an die Software die im Laufe des Projektes umgesetzt werden sollen. Die Features sind dabei in Pflicht- und optionale Funktionalitäten eingeteilt und absteigend nach ihrer Priorität geordnet.

Die Implementierung der Features erfolgt nach absteigender Priorität, denn einige Features hängen von anderen Features ab. Diese Abhängigkeit wurde in der Priorisierung der Features berücksichtigt.

Pflicht-Funktionalitäten

1. Manuelles Eingeben und Korrigieren von Rezepten
2. Erstellen von Rezeptvorschlägen nach Eingabe von vorhandenen Zutaten
3. Rezepte von Websites importieren
4. Verwaltung ähnlicher Zutaten
5. Einbeziehen ähnlicher Zutaten bei der Suche
6. Ausschließen und Ersetzen von bestimmten Zutaten
7. Rezepte als ausdrucksbares Dokument exportieren
8. Bewertung von Rezepten durch Benutzer
9. Statistische Auswertungen (am meisten gekocht, am besten bewertet, etc.)

Optionale Funktionalitäten

1. Anzeigen von Bildern zu den Rezepten
2. Virtueller Kühlschrank (merken von vorhandenen Zutaten zwischen Programmaufrufen)
3. Liste der fehlenden Zutaten als Einkaufsliste anzeigen und exportieren
4. Themen-Suche (z.B. asiatisch)
5. Zutaten-Menge auf eine bestimmte Personenanzahl umrechnen
6. Kalorienberechnung der Speisen + Suchkriterium Kaloriengehalt
7. Interface für Barcode-Scanner (Aktualisieren des virtuellen Kühlschranks)

Domänenmodell

Die Grundfunktionalität ist in drei Bereiche unterteilt:

1. die Verwaltung der Rezepte
2. die Suche nach Rezepten basierend auf vorhandenen Zutaten oder Stichworten
3. die weitere Verarbeitung der Rezepte bzw. der virtuelle Kühlschrank

Die Eingabe und das Downloaden von Rezepten aus dem Internet läuft über den Rezept-Downloader ab, der bestimmte Webseiten mit Rezepten parsen kann und in der lokalen Rezepte-Datenbank in einem einheitlichen Format, das von den anderen Programmteilen verstanden wird, speichert.

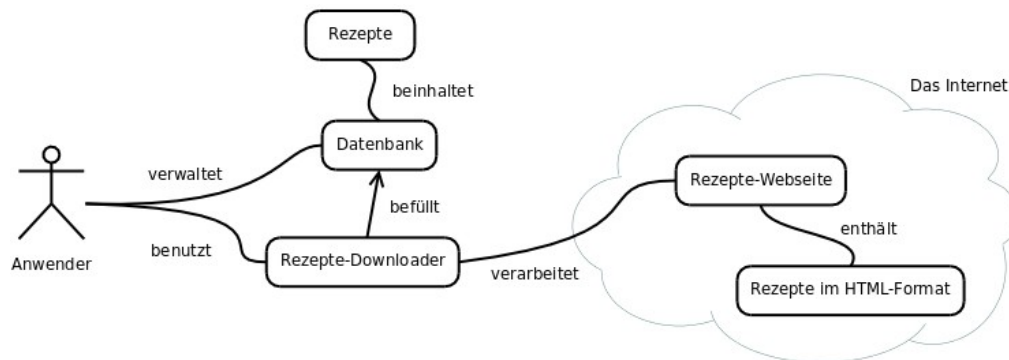


Abbildung 1: Domänenmodell "Rezepte verwalten"

Außerdem ermöglicht das Programm die Suche nach Rezepten mit zusätzlichen Einschränkungen.

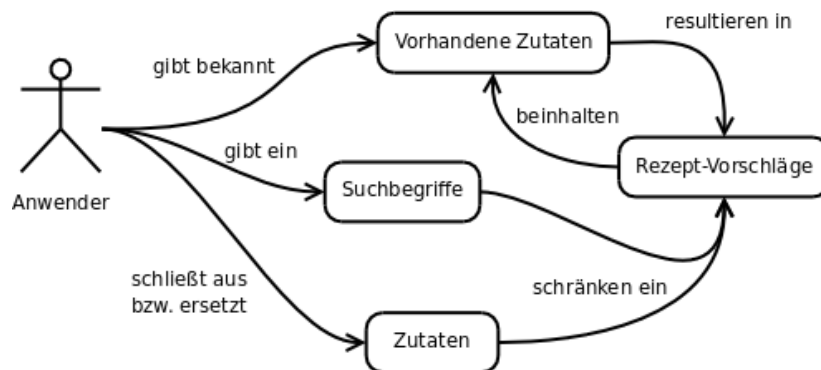


Abbildung 2: Domänenmodell "Rezepte suchen"

Nachdem der Anwender ein Rezept ausgewählt hat, kann er dieses im Weiteren noch verarbeiten (bewerten, Zutaten auf andere Personenanzahl umrechnen, exportieren). Außerdem hat der Anwender die Möglichkeit einen virtuellen Kühlschrank zu verwalten, mit dem er in Verbindung mit dem aktuellen Rezept auch eine Einkaufsliste erstellen kann.

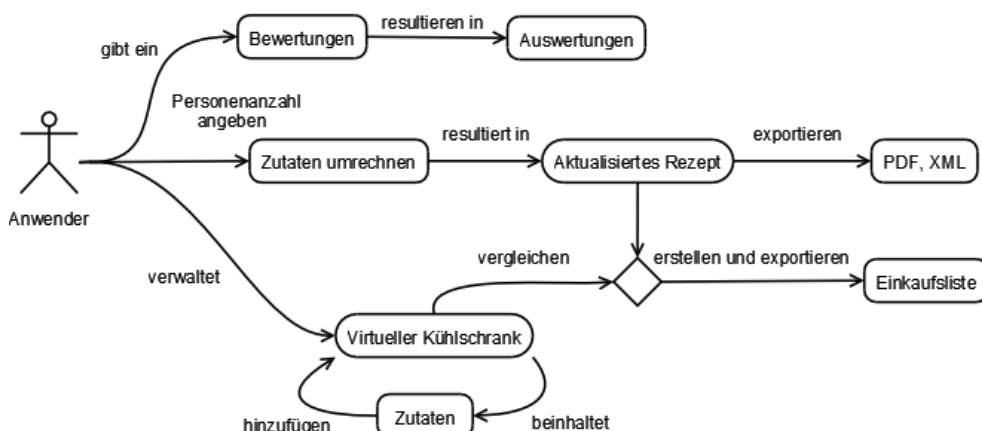


Abbildung 3: Domänenmodell "virtueller Kühlschrank"

Funktionale Anforderungen

Der folgende Abschnitt listet die funktionalen Anforderungen an die zu erstellende Software. Es wird jeweils der Ablauf bei erfolgreicher bzw (sofern zutreffend) bei negativer Ausführung beschrieben.

Liste der Anwendungsfälle

Die folgende Liste spiegelt den Zusammenhang von Anwendungsfall und Feature, geordnet nach deren Priorität, wider.

Anwendungsfall	Priorität
Erstellen von Rezepten	1
Korrigieren von Rezepten	1
Anzeigen von Rezepten	1
Löschen von Rezepten	1
Suchkriterien eingeben und Rezept suchen	2
Rezept aus Suchergebnissen auswählen und anzeigen	2
Rezepte Website auswählen und importieren	3
Eingeben von ähnlichen Zutaten	4
Bearbeiten von ähnlichen Zutaten	4
Löschen von ähnlichen Zutaten	4
Eingeben von zu ersetzenden Zutaten	5
Bearbeiten von zu ersetzenden Zutaten	5
Löschen von zu ersetzenden Zutaten	5
Eingeben von auszuschließenden Zutaten	6
Bearbeiten von auszuschließenden Zutaten	6
Löschen von auszuschließenden Zutaten	6
Angezeigtes Rezept exportieren	7
Rezept bewerten	8
Rezept als "gekocht" markieren	8
Auswertung der meist gekochten Rezepten	9
Auswertung der am besten bewerteten Rezepte	9
Rezept-Bild aus Datei laden	opt 1
Eintragen von Zutaten in den virtuellen Kühlschrank	opt 2
Bearbeiten von Zutaten im virtuellen Kühlschrank	opt 2
Herausnehmen von Zutaten aus dem virtuellen Kühlschrank	opt 2
Einkaufsliste berechnen und anzeigen	opt 3
Angezeigte Einkaufsliste exportieren	opt 3
Einkaufsliste bearbeiten	opt 3
Thema eingeben und suchen	opt 4
Eingabe Anzahl Personen, Zutatenmenge umrechnen	opt 5
Kalorien der Speisen berechnen	opt 6
Aktualisieren der gescannten Einkäufe	opt 7

Anwendungsfalldiagramme

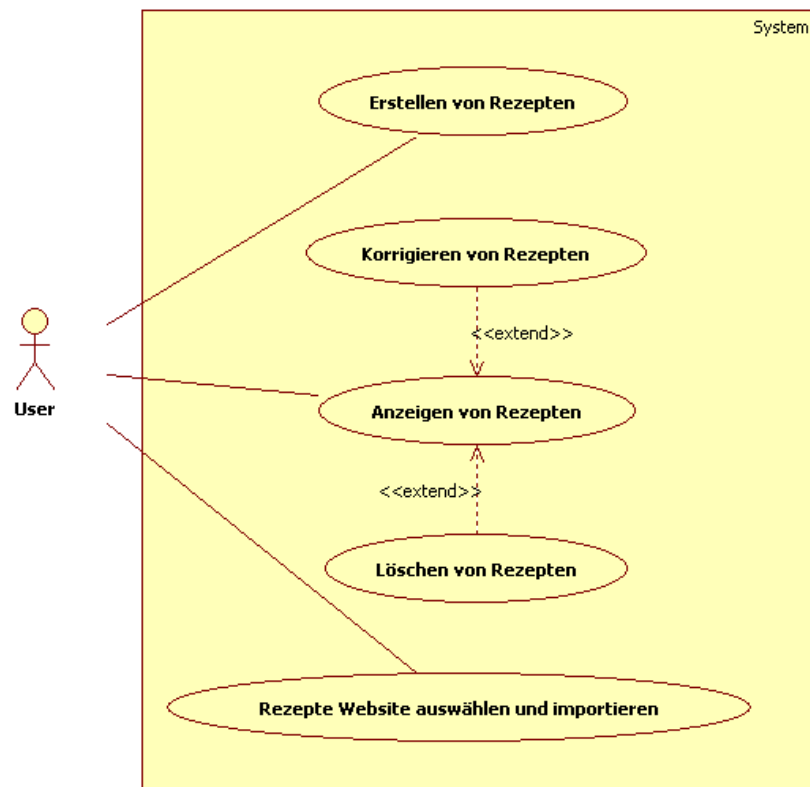


Abbildung 4: Anwendungsfalldiagramm "Manuelles Eingeben und Korrigieren von Rezepten" mit "Rezepte von Websites importieren"

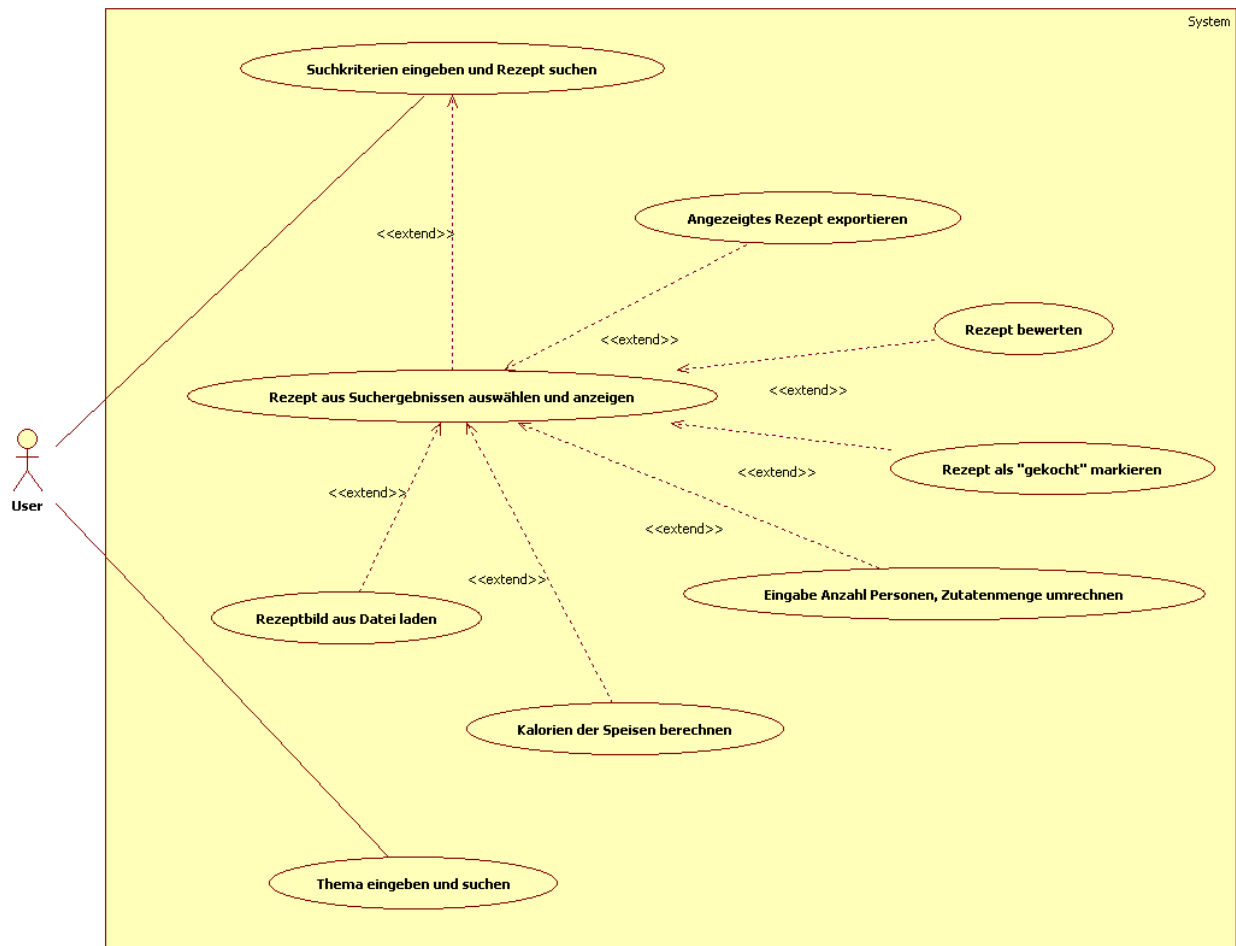


Abbildung 5: Anwendungsfalldiagramm "Erstellen von Rezeptvorschlägen nach Eingabe von vorhandenen Zutaten"

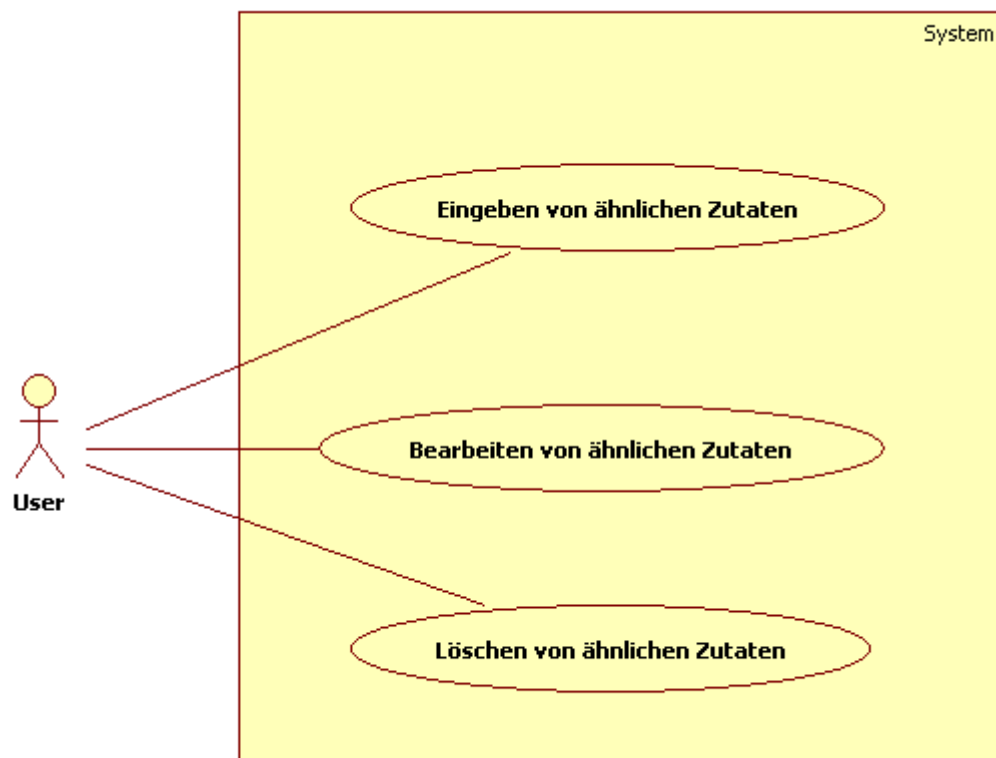


Abbildung 6: Anwendungsfalldiagramm "Verwaltung ähnlicher Zutaten"

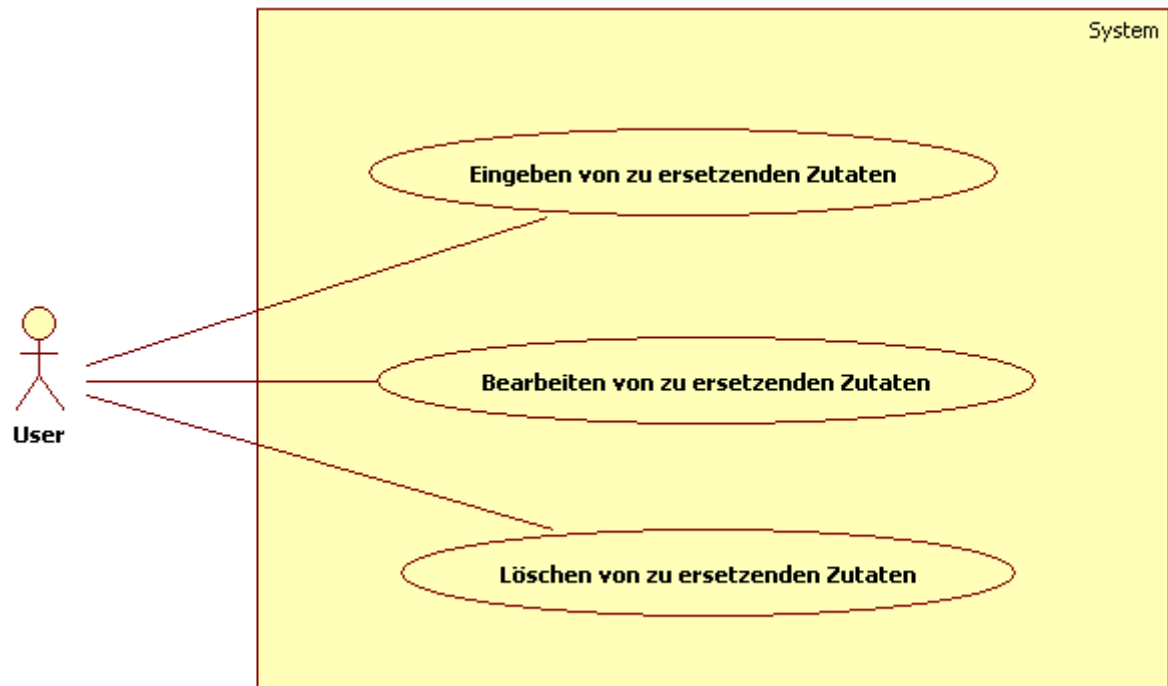


Abbildung 7: Anwendungsfalldiagramm "Einbeziehen ähnlicher Zutaten bei der Suche"

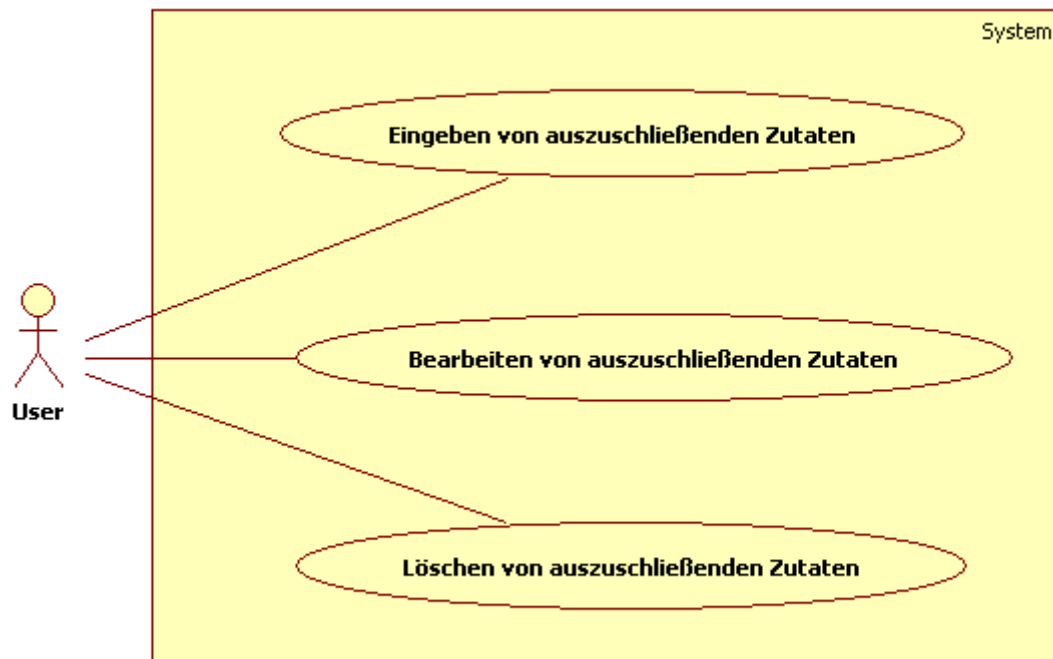


Abbildung 8: Anwendungsfalldiagramm "Ausschließen und Ersetzen von bestimmten Zutaten"

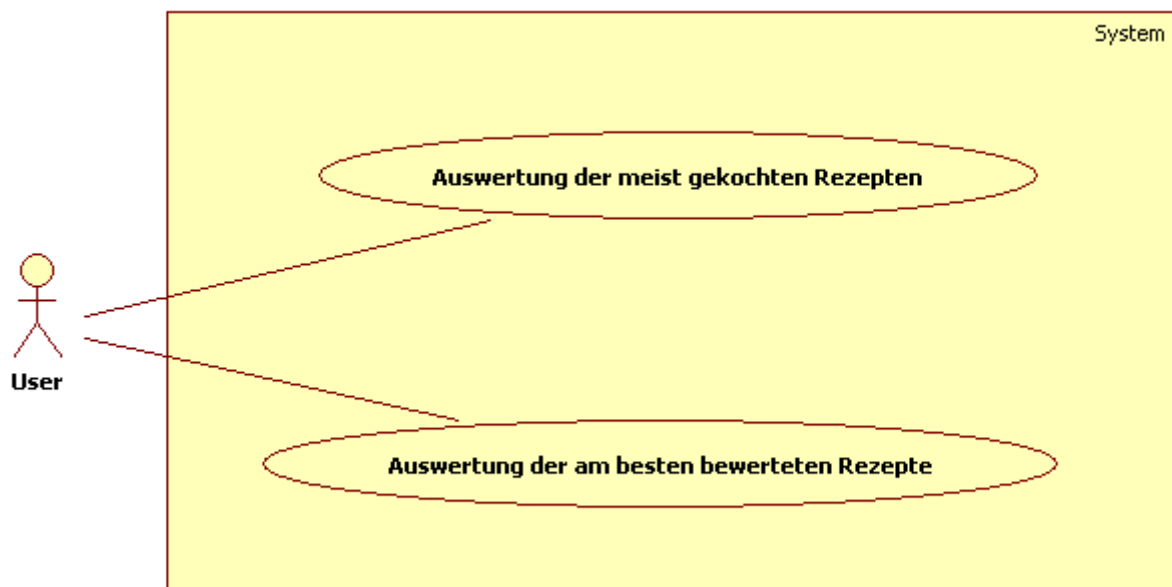


Abbildung 9: Anwendungsfalldiagramm "Statistische Auswertungen (am meisten gekocht, am besten bewertet, etc.)"

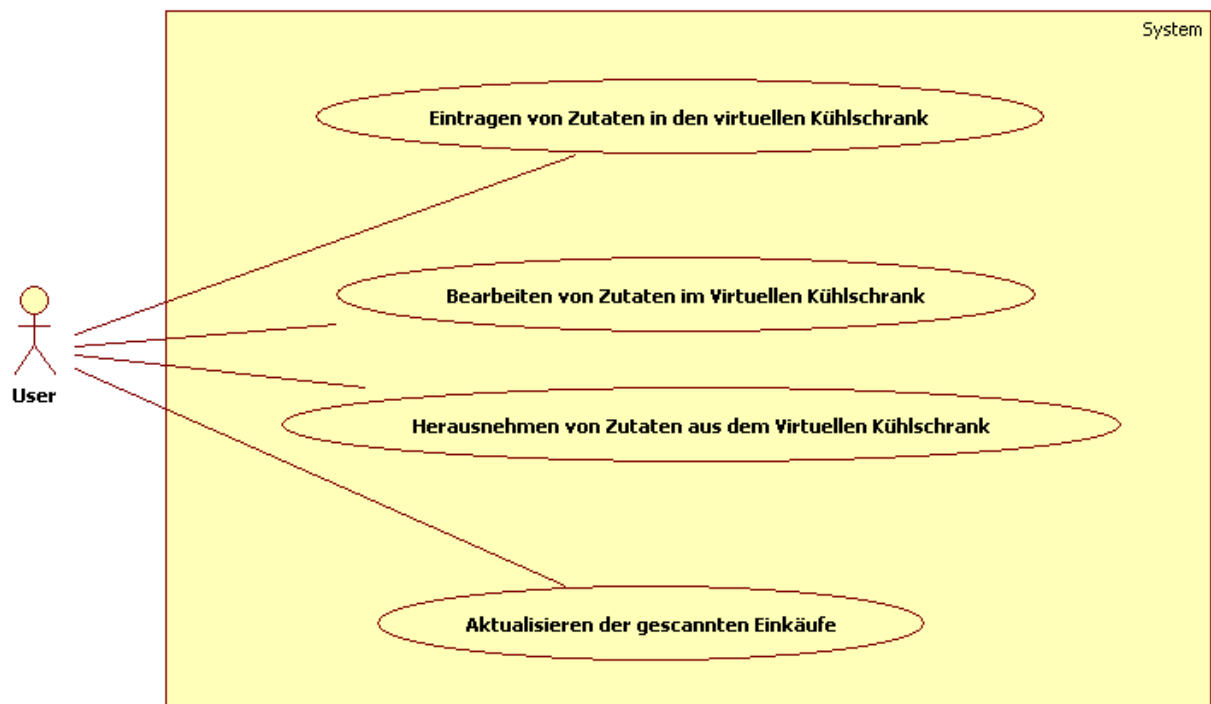


Abbildung 10: Anwendungsfalldiagramm "Virtueller Kühlschrank"

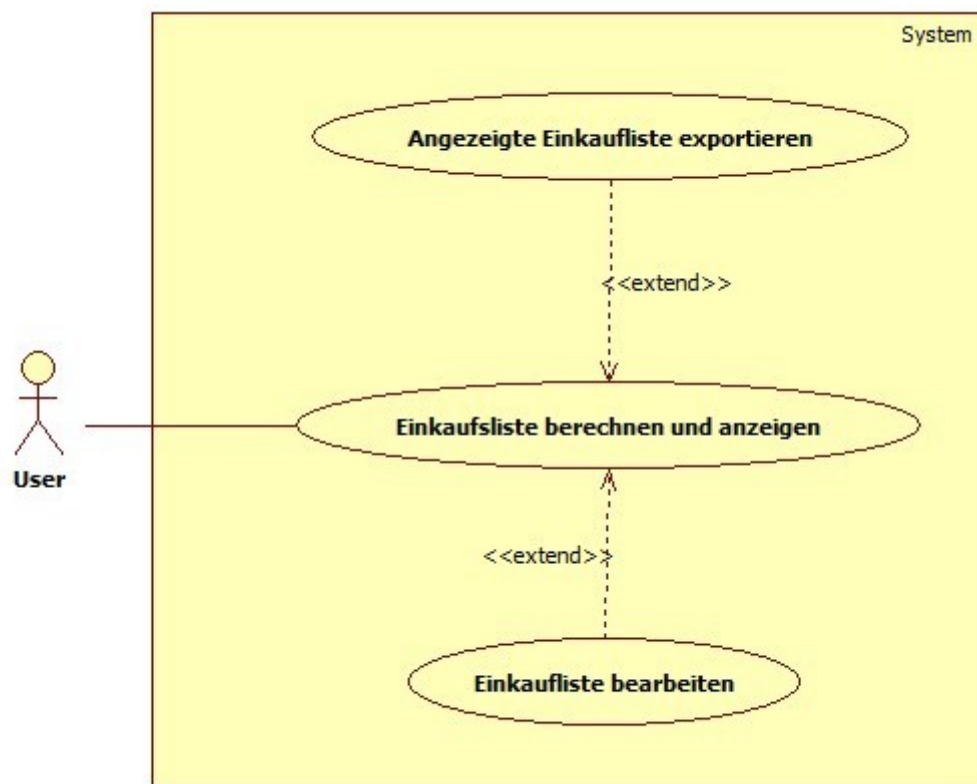


Abbildung 11: Anwendungsfalldiagramm "Einkaufsliste"

Anwendungsfallbeschreibungen

Rezept bearbeiten

Der User legt ein neues Rezept an oder bearbeitet bzw löscht ein bestehendes.

- **Ablauf**
 - Zum Bearbeiten und Löschen muss zuerst ein Rezept aus der Liste von gespeicherten Rezepten ausgewählt werden.
 - Beim Speichern bzw Erstellen eines Rezepts gibt der Benutzer Rezeptname, Zutaten und Zubereitungsinformationen in einem Formular ein. Das Programm speichert anschließend die Daten in der Datenbank.
 - Zum Löschen wählt der Benutzer die Löschen-Aktion für die ausgewählten Rezepte aus.
- **Fehlersituationen:** Falls erforderliche Informationen nicht angegeben werden oder die Änderungen an der Datenbank nicht durchgeführt werden konnten, erfolgt eine Fehlermeldung.
- **Nachbedingung:** Die Änderungen an der Datenbank wurden vorgenommen.

Suchkriterien eingeben und Rezept suchen

Der Benutzer sucht nach einem Rezept, indem er die vorhandenen bzw gewünschten Zutaten eingibt.

- **Ablauf**
 - Der Benutzer gibt die gewünschten Zutaten ein.
 - Das Programm durchsucht anhand dieser Eingabe die Datenbank und berücksichtigt dabei ähnliche, ausgeschlossene und alternative Zutaten.
 - Das Programm zeigt danach eine Liste aller gefundenen Rezepte an.
- **Nachbedingung** Die gefundenen Rezepte werden ausgegeben.

Rezept aus Suchergebnissen auswählen und anzeigen

Der Benutzer wählt ein Rezept aus den Suchergebnissen und zeigt es an.

- **Vorbedingung:** Suchergebnisse nach Rezeptsuche vorhanden.
- **Ablauf**
 - Der Benutzer wählt ein Rezept aus den Suchergebnissen aus und zeigt es an.
 - Das Programm zeigt dem Benutzer die notwendigen Zutaten, ggf ersetzt durch Alternativ-Zutaten sowie die Zubereitungsanweisungen an.

Rezepte-Website auswählen und importieren

Der Benutzer wählt eine unterstützte Rezept-Website und kann Rezepte von dieser importieren.

- **Vorbedingung:** Es sind Parser-Skripte für Websites installiert.
- **Ablauf**
 - Dem Benutzer wird eine Liste der vorhandenen Parser-Skripte (und ihrer unterstützten Websites) angezeigt. Aus diesen wählt er eines aus.
 - Das Programm zeigt dem Benutzer eine Liste aller gefundenen Rezepte auf der gewählten Website an.
 - Der Benutzer wählt die Rezepte, die importiert werden sollen.
 - Das Programm importiert die gewählten Rezepte und überträgt sie in die Datenbank.
- **Fehlersituationen**

- Die Verbindung zur Website kann nicht hergestellt werden.
- Die Rezepte können nicht geparsed bzw verarbeitet werden.
- **Nachbedingungen**
 - Die vom Benutzer gewählten Rezepte befinden sich in der Datenbank.

Bearbeiten von ähnlichen Zutaten

Der Benutzer legt fest, welche Zutaten vom Programm als ähnlich angesehen werden. Rezepte mit Zutaten, die den Suchkriterien ähnlich sind, werden in den Suchergebnissen berücksichtigt.

- **Ablauf**
 - Der Benutzer wählt zwei vorhandene Zutaten, bzw gibt sie ein.
 - Das Programm markiert die gewählten Zutaten in der Datenbank als ähnlich.
 - Zum Löschen wählt der Benutzer aus der Liste aller Ähnlichkeiten einen Eintrag aus und löscht ihn.
- **Nachbedingungen**
 - Die Änderungen wurden in die Datenbank übernommen.
 - Bei zukünftigen Rezept-Suchen werden beide Zutaten als ähnlich behandelt.

Bearbeiten von zu ersetzenden Zutaten

Der Benutzer legt fest, welche Zutaten in den Rezepten durch andere ersetzt werden sollen. Rezepte mit zu ersetzenden Zutaten werden zwar weiterhin gefunden, jedoch werden die entsprechenden Zutaten bei der Anzeige durch ihre Alternativen ersetzt.

- **Ablauf**
 - Der Benutzer wählt zuerst die Zutat, die er in Ihren Rezepten durch eine andere ersetzen möchte.
 - Danach wählt er die Zutat, mit der die zuvor gewählte ersetzt werden soll.
 - Das Programm speichert den Eintrag in der Datenbank.
 - Zum Löschen wählt der Benutzer aus der Liste aller ersetzten Zutaten einen Eintrag aus und löscht ihn.
- **Nachbedingungen**
 - Die Änderungen wurden in die Datenbank übernommen.
 - Bei zukünftigen Rezept-Suchen wird statt der ersetzten Zutat immer die ersetzende verwendet.

Bearbeiten von auszuschließenden Zutaten

Der Benutzer legt fest, welche Zutaten nicht in den Rezepten vorkommen sollen. Rezepte die solche Zutaten erhalten, werden nicht mehr in den Suchergebnissen angezeigt.

- **Ablauf**
 - Der Benutzer wählt eine vorhandene Zutat bzw gibt eine neue ein.
 - Das Programm speichert diese Zutat als ausgeschlossen in der Datenbank.
 - Zum Löschen wählt der Benutzer aus der Liste aller auszuschließenden Zutaten eine aus und löscht diese.
- **Nachbedingungen**
 - Die Änderungen wurden in die Datenbank übernommen.
 - Bei zukünftigen Rezept-Suchen werden keine Rezepte mehr angezeigt, die eine auszuschließende Zutat enthalten.

Angezeigtes Rezept exportieren

Der Benutzer exportiert das angezeigte Rezept in ein anderes Dateiformat.

- **Vorbedingungen**
 - Der Benutzer hat ein Rezept aus den Suchergebnissen angezeigt.
- **Ablauf**
 - Dem Benutzer wird eine Auswahl der zur Verfügung stehenden Formate angezeigt.
 - Der Benutzer gibt einen Namen für die zu erstellende Datei ein.
 - Das aus den Suchergebnissen angezeigte Rezept wird vom Programm in das gewählte Format exportiert.
- **Nachbedingungen**
 - Das angezeigte Rezept ist als Datei im gewählten Format vorhanden.

Rezept bewerten

Der Benutzer bewertet das aktuell angezeigte Rezept.

- **Vorbedingungen**
 - Der Benutzer hat ein Rezept aus den Suchergebnissen angezeigt.
- **Ablauf**
 - Der Benutzer vergibt eine Bewertung für das angezeigte Rezept.
 - Das Programm speichert die Bewertung für das Rezept in der Datenbank.
- **Nachbedingungen**
 - Die Bewertung ist in der Datenbank gespeichert.

Rezept als "gekocht" markieren

Der Benutzer markiert das aktuell angezeigte Rezept als "gekocht".

- **Vorbedingungen**
 - Der Benutzer hat ein Rezept aus den Suchergebnissen angezeigt.
- **Ablauf**
 - Der Benutzer markiert das angezeigte Rezept als "gekocht".
 - Das Programm speichert die Markierung für das Rezept in der Datenbank.
- **Nachbedingungen**
 - Die Markierung ist in der Datenbank gespeichert.

Auswertung der meist-gekochten Rezepte

Der Benutzer zeigt die Liste mit den meist-gekochten Rezepten an.

- **Vorbedingung**
 - Der Benutzer hat bereits Rezepte als "gekocht" markiert.
- **Ablauf**
 - Der Benutzer öffnet die Auswertung der meist-gekochten Rezepte.
 - Das Programm erstellt eine Rezept-Liste anhand der Markierungen und zeigt diese an.
 - Der Benutzer kann Rezepte aus der angezeigten Liste anzeigen.
- **Nachbedingungen**
 - Dem Benutzer wird eine Liste der meist-gekochten Rezepte angezeigt, aus der er einzelne auswählen und anzeigen kann.

Auswertung der am besten bewerteten Rezepte

Der Benutzer zeigt die Liste mit am besten bewerteten Rezepten an.

- **Vorbedingung**
 - Der Benutzer hat bereits Rezepte bewertet.
- **Ablauf**

- Der Benutzer öffnet die Auswertung der am besten bewerteten Rezepte.
- Das Programm erstellt eine Rezept-Liste anhand der Bewertungen und zeigt diese an.
- Der Benutzer kann Rezepte aus der angezeigten Liste anzeigen.
- **Nachbedingungen**
 - Dem Benutzer wird eine Liste der am besten bewerteten Rezepte angezeigt, aus der er einzelne auswählen und anzeigen kann.

Rezeptbild aus Datei laden

Der Benutzer weist einem Rezept ein Bild aus einer Datei zu.

- **Vorbedingungen**
 - Der Benutzer hat ein Rezept aus den Suchergebnissen angezeigt.
- **Ablauf**
 - Der Benutzer öffnet die Datei mit dem Rezeptbild.
 - Das Programm überträgt das Bild in die Datenbank und verbindet es mit dem aktuell angezeigten Rezept.
- **Nachbedingungen**
 - Das Bild ist in der Datenbank gespeichert und mit dem aktuell angezeigten Rezept verknüpft.

Bearbeiten von Zutaten im Virtuellen Kühlschrank

Der Benutzer gibt neue Zutaten in den Kühlschrank bzw nicht vorhandenen Zutaten heraus.

- **Vorbedingungen**
 - Zum Entnehmen von Zutaten müssen bereits welche im Kühlschrank enthalten sein.
- **Ablauf**
 - Der Benutzer wählt Zutaten aus, die in den virtuellen Kühlschrank eingefügt werden solln.
 - Das Programm speichert die vorhandenen Zutaten und berücksichtigt sie bei späteren Suchen.
 - Zum Löschen wählt der Benutzer im Kühlschrank vorhandene Zutaten aus und löscht sie.
- **Nachbedingungen**
 - Die Änderungen wurden in die Datenbank übertragen.

Einkaufsliste berechnen und anzeigen

Der Benutzer lässt sich eine Einkaufsliste erstellen.

- **Vorbedingungen**
 - Art und Menge der vorhandenen Zutaten wurden eingegeben
 - Der Benutzer hat ein Rezept aus den Suchergebnissen angezeigt.
- **Ablauf**
 - Der Benutzer lässt sich eine Einkaufsliste erstellen.
 - Aus den vorhandenen Zutaten (im virtuellen Kühlschrank) und den benötigten Zutaten für das angezeigte Rezept wird eine Einkaufsliste erstellt.
- **Nachbedingungen**
 - Dem Benutzer wird eine Einkaufsliste mit den fehlenden Zutaten angezeigt.

Angezeigte Einkaufsliste exportieren

Der Benutzer exportiert die angezeigte Einkaufsliste.

- **Vorbedingungen**
 - Eine Einkaufsliste wurde erstellt und angezeigt.
- **Ablauf**
 - Dem Benutzer wird eine Auswahl der zur Verfügung stehenden Formate angezeigt.
 - Der Benutzer gibt einen Name für die zu erstellende Datei ein.
 - Die angezeigte Einkaufsliste wird vom Programm in das gewählte Format exportiert.
- **Nachbedingungen**
 - Die angezeigte Einkaufsliste ist als Datei im gewählten Format vorhanden.

Einkaufsliste bearbeiten

Der Benutzer passt die erstellte Einkaufsliste an.

- **Vorbedingungen**
 - Eine Einkaufsliste wurde erstellt und angezeigt.
- **Ablauf**
 - Der Benutzer fügt neue Einträge in die Einkaufsliste ein bzw bearbeitet oder löscht vorhandene.
 - Der Benutzer exportiert die angepasste Einkaufsliste.

Thema eingeben und suchen

Der Benutzer sucht Rezepte nach einem bestimmten Thema z.B. "asiatisch" oder "italienisch".

- **Vorbedingungen**
 - Es sind Rezepte vorhanden, zu denen ein Thema gespeichert ist.
- **Ablauf**
 - Der Benutzer gibt ein Thema ein.
 - Das Programm erstellt eine Liste mit allen passenden Rezepten.

Eingabe Anzahl Personen, Zutatenmenge umrechnen

Der Benutzer rechnet die Zutatenmengen anhand der Portionsanzahl um.

- **Vorbedingungen**
 - Der Benutzer hat ein Rezept aus den Suchergebnissen angezeigt.
- **Ablauf**
 - Der Benutzer gibt die Anzahl der gewünschten Personen bzw Portionen ein
 - Das Programm rechnet die Mengen des gespeicherten Rezepts für die angegebene Personen- bzw Portionsanzahl um.
- **Nachbedingungen**
 - Im angezeigten Rezept sind die Mengen an die angegebene Personen- bzw Portionsanzahl angepasst.

Kalorien der Speisen berechnen

Der Benutzer lässt sich die Kalorien des aktuell angezeigten Rezepts berechnen.

- **Vorbedingungen**
 - Zu den Zutaten sind Kalorien-Informationen gespeichert.
 - Der Benutzer hat ein Rezept aus den Suchergebnissen angezeigt.
- **Ablauf**
 - Der Benutzer startet die Berechnung der Gesamt-Kalorien des aktuell angezeigten Rezepts.
- **Nachbedingung**

- Die Gesamt-Kalorien des aktuell angezeigten Rezepts werden angezeigt.

Aktualisieren der gescannten Einkäufe

Der Benutzer lässt die Zutaten im virtuellen Kühlschrank durch Scannen von gekauften Produkten aktualisieren.

- **Vorbedingungen**
 - Der Benutzer verfügt über eine Möglichkeit zum Scannen von Barcodes.
- **Ablauf**
 - Der Benutzer startet die Funktion zum Scannen von Produkten.
 - Der Benutzer scannt die Barcodes der gekauften Produkte.
 - Die Informationen im virtuellen Kühlschrank werden aktualisiert.
- **Nachbedingungen**
 - Die Art und Menge der vorhandenen Zutaten wird im virtuellen Kühlschrank aktualisiert.

Informationswesen

Die Kommunikation zwischen Auftraggeber erfolgt über den Projektleiter. Dieser verarbeitet die Informationen und gibt wichtige Informationen über die Projekt-Mailingliste bekannt.

Meeting Notes werden im Trac-Wiki geschrieben und verarbeitet, da uns dies ermöglicht, Tickets zu im Meeting beschlossenen Tasks direkt in den Meeting Notes zu verlinken, und so einen aktuellen Überblick zu behalten.

Besprechungen werden vereinbart, indem ein Projektmitglied einen Termin im **Doodle**-Webservice einträgt, und den Termin und die URL der Doodle- Umfrage per Mailingliste bekannt gibt. Das letzte Projektmitglied, das sich im Doodle einträgt ist dafür zuständig, das Ende der Umfrage und den vereinbarten Termin den anderen Gruppenmitgliedern über die Mailingliste bekannt zu geben.

Kurzfristig koordinierte Treffen können auch im IRC-Chat (Channel #sepm auf Freenode, wenn nicht anders angegeben) vereinbart werden. Für IRC- Meetings werden ebenso Meeting Notes erstellt. Im Normalfall sollten alle Treffen aber wegen der erhöhten Aufmerksamkeit *in Person* erfolgen.

Die **Stundenaufzeichnung** erfolgt über das Ticket-System im Trac.

Die **Projektdokumentation** wird in **Maven Site** abgelegt.

Arbeitsstruktur und Rollenverteilung

Die Rollenverteilung findet sich auf der Titelseite dieses Dokuments.

Aufgaben werden während des Projektverlaufs den Rollen entsprechend vergeben. Allgemeine Features werden unabhängig von der Rolle zeitnah in den regelmäßigen Meetings zugeteilt.

Allgemeine Zuständigkeiten aller Projektmitglieder

- Mitarbeit in möglichst allen Bereichen des Projekts (Organisation, Code, Dokumentation), um den Lerneffekt zu maximieren und sich entsprechend gut für das Kontrollgespräch am Ende des Semesters vorzubereiten
- Regelmäßiges Durchlesen und Bearbeiten von E-Mails (zumindest einmal täglich)
- Pünktliches Erscheinen zu Meetings; Anwesenheit bei Meetings - Fernbleiben nur bei vorheriger, begründeter Notiz an die anderen Projektmitglieder
- Bearbeiten und Abschließen von zugewiesenen Tasks bis zur vereinbarten Deadline

- Einhaltung von Coding-Standards, der Netiquette und der deutschen Rechtschreibung in allen Bereichen des Projekts; die Coding-Standards werden im Trac Wiki veröffentlicht und gegebenenfalls aktualisiert
- Tickets im Trac immer auf dem aktuellen Stand halten (Status, Stunden, ...)

Zuständigkeiten des Teamkoordinators

- Kommunikation zwischen dem Tutor und dem Projektteam
- Tasks erstellen und Projektmitgliedern zuweisen
- Verwaltung und Abschluss von Milestones
- Wichtige Informationen des Tutors an die Mailingliste weiterleiten
- Überwachung des Projektverlaufs und Motivation der Projektmitglieder
- Dokumentation von nicht eingehaltenen Deadlines und Weiterleitung dieser Information an den Tutor
- Ansprechpartner für etwaige Fragen und Probleme bei Projektmitgliedern

Zuständigkeiten des technischen Architekten

- Ständiger Überblick über die Architektur des Projekts
- Erkennen von Verständnisproblemen über die Architektur und Beseitigung dieser
- Ansprechpartner für allgemeine technische Fragen und Tool-Troubleshooting
- Pflege des SVN-Repository, um den Code geordnet und strukturiert zu halten
- Koordination von Refactoring-Tätigkeiten, Branching und Tagging
- Unterstützung des Teamkoordinators bei der Durchsicht der Tickets

Zuständigkeiten des Dokumentenbeauftragten

- Regelmäßige Kontrolle aller abzugebenden Artefakte aufgrund der Artefaktenliste
- Koordination der korrekten Erstellung von Meeting Notes (Folge: Hat die Kompetenz, Projektmitglieder als Protokollanten bei Meetings einzuteilen)
- Ansprechpartner für Fragen zu Javadoc, Maven und ähnlichen Technologien
- Verantwortlich für die Konfiguration und Wartung unserer Maven Site-Instanz
- Absegnen (Kontrolle!) von Dokumenten, die zur Abgabe bereit stehen

Zuständigkeiten des Testbeauftragten

- Regelmäßige Kontrolle der Vollständigkeit (Coverage) von Unittests
- Ansprechpartner für Fragen zu JUnit, Unittests allgemein und Mocking
- Koordination der Erstellung von Unit-Tests und Testplänen (Folge: Hat die Kompetenz, Tickets für Projektmitglieder zu erstellen, wenn Testfälle fehlen)
- Verantwortlich für die Wartung der Ordner- und Codestruktur der Tests im SVN-Repository

Zuständigkeiten des User Interface Designers

- Durchführung von Usability Tests mit Testpersonen (und Organisation dieser), falls diese vom Projektteam gewünscht werden (nach Meeting-Beschluss)
- Ansprechpartner für Fragen zum Design aller User-sichtbaren Programmteile
- Verantwortlich für die Erhaltung einer guten Usability (Folge: Hat die Kompetenz, Tickets für Projektmitglieder zu erstellen, wenn die UI Fehler aufweist oder inkonsistent ist)
- Erstellung von attraktiven UI-Screenshots (mit sinnvollen Daten in Eingabemasken) für Dokumentation und UI-Reviews
- Verantwortlich für die Verständlichkeit des Benutzerhandbuchs

Zuständigkeiten des Datenbankbeauftragten

- Ständiger Überblick über die aktuelle Datenbankstruktur (Schema, Daten) und beheben von Problemen
- Ansprechpartner für Fragen zu HSQLDB, JDBC, dem ORM-Mapper und Entity-Objekten
- Verantwortlich für das Vorhandensein von sinnvollen Testdaten (Folge: Hat die Kompetenz, Tickets für Projektmitglieder zu erstellen, um die Eingabe von Testdaten in die Datenbank anzufordern)

Projektplan

Das Projekt ist in einwöchige Sprints aufgeteilt, deren Resultat mit Ausnahme des ersten Sprints immer ein Software-Release des Prototypen ist. Sprints finden immer von Donnerstag bis Donnerstag statt und beginnen und enden mit dem internen Review zwischen Projektteam und Tutor.

Die ersten beiden Sprints haben noch den Schwerpunkt der Dokumentation und Planung, in den weiteren Sprints liegt die Priorität auf die agile Weiterentwicklung des Prototypen mit dazugehöriger Erweiterung und Korrektur der Dokumentation.

Sprint	Beschreibung/Artefakte	Beginn	Ende
Paper Taco	Work Breakdown Structure, Anwendungsfallbeschreibung Komponentendiagramm, UI-Skizze (Paper Prototype) Vorläufiges ER-Diagramm	30.04.2009	07.05.2009
Taco 0.1	Deployment-Diagramm, GANTT-Diagramm, UI Beschreibung Datenbankbeschreibung (ER-Diagramm und Attribute) Lauffähiger Prototyp (Feature 1 fertig implementiert)	07.05.2009	14.05.2009
Taco 0.2	Lauffähiger Prototyp (Features 1, 2 fertig; 3 teilweise) Präsentation zum MR2 (20.5.); Testplan und Testfälle	14.05.2009	21.05.2009
Taco 0.4	Lauffähiger Prototyp (Features 1 bis 3 implementiert) Klassendiagramm; Testplan und Testfälle fertig Fehler im Programm (MR2) beheben	21.05.2009	28.05.2009
Taco 0.6	Lauffähiger Prototyp (Features 1 bis 6 implementiert) Installations- und Benutzerhandbuch	28.05.2009	04.06.2009
Taco 0.8	Lauffähiger Prototyp (Features 1 bis 8 implementiert) Projektendbericht; Testbericht	04.06.2009	11.06.2009
Taco 0.9	Lauffähiger Prototyp (Features 1 bis 9 implementiert) Präsentation zum MR3 (Juni); Feinschliff der Must- Haves	11.06.2009	18.06.2009
Taco 1.0 RC	Lauffähiger Release Candidate; Optionale Features 1 bis 4 implementiert	18.06.2009	25.06.2009
Taco 1.0	Lauffähiges Release allgemein verfügbar	25.06.2009	28.06.2009

Risikoabschätzung

Im folgenden sind Risiken gelistet, die während der Projektlaufzeit eintreten können. Zusätzlich sind mögliche Gegenmaßnahmen zu den einzelnen Risiken beschrieben. Die Gegenmaßnahmen, die bei Eintreten eines Risikos gewählt werden, hängen jedoch immer von der konkreten Situation ab.

1. Ausfall eines Projektmitglieds

Ein Projektmitglied kann, zB bedingt durch eine Krankheit, längere Zeit nicht am Projekt mitarbeiten bzw scheidet komplett aus dem Projektteam aus.

Gegenmaßnahmen: Sammeln der bereits geleisteten und noch ausstehenden Arbeit; Besprechung und Aufteilung der Aufgaben innerhalb der Gruppe. Zuteilung der Rolle zu einem anderen Gruppenmitglied.

Art	Allgemeines Risiko
Typisierung	immer
Priorität	Hoch
Eintrittswahrscheinlichkeit	Niedrig
Folgewirkung	2., 6.
Verantwortlichkeit	Perl, Kögl

2. Rechtzeitige Fertigstellung gefährdet

Die rechtzeitige Fertigstellung des Projekts ist gefährdet. Mögliche Ursachen könnten zu hohe Anforderungen, zu geringe zeitliche Ressourcen, Ausfall von Team-Mitgliedern, etc sein.

Gegenmaßnahmen: Reduktion von Features anhand ihrer Priorisierung (abhängig von aktuellem Fortschritt)

Art	allgemeines Risiko
Typisierung	Entwicklung, Management
Priorität	Hoch
Eintrittswahrscheinlichkeit	Niedrig
Verantwortlichkeit	Perl, Kögl

3. Unvorhergesehene Änderungen an Rezepte-Websites

Unvorhergesehene Änderungen an Rezepte-Websites können bereits implementierte Rezepte-Parser unbrauchbar machen.

vorbeugende Gegenmaßnahmen:

- Modularisierung des Parsens; jede Website erhält ihren eigenen Parser. Parser können als Plugins ergänzt bzw ausgetauscht werden
- Parser wird in Python mit entsprechenden Libraries implementiert, was einfachere Änderungen als eine Java-Implementierung zulässt

Art	Projekt-spez. Risiko
Typisierung	Architektur
Priorität	Mittel

Eintrittswahrscheinlichkeit	Mittel
Verantwortlichkeit	Kögl, Perl

4. Notwendige Libraries nicht verfügbar

Während des Projektverlaufs stellt sich heraus, dass für ein bestimmtes Feature notwendige Libraries nicht (frei) verfügbar sind.

Gegenmaßnahmen:

- sofern der Aufwand vertretbar ist, können die Funktionalitäten selbst entwickelt, bzw andere Libraries adaptiert werden
- ist dies nicht möglich, muss das Feature so weit wie notwendig reduziert werden

Art	Allgemeines Risiko
Typisierung	Planung, Entwicklung
Priorität	Hoch
Eintrittswahrscheinlichkeit	Niedrig
Folgewirkung	2.
Verantwortlichkeit	Kögl, Perl

5. Unzureichende Design-Entscheidungen

Während der Implementierung stellt sich eine Design-Entscheidung als unzureichend heraus.

Gegenmaßnahmen:

- Design-Entscheidungen werden so früh wie möglich mit allen Projektmitgliedern besprochen, die die Entscheidung bezüglich Ihrer Rolle beurteilen
- Modulares Design, damit die Auswirkungen falscher Design-Entscheidungen möglichst klein gehalten werden

Art	Allgemeines Risiko
Typisierung	Planung, Entwicklung
Priorität	Mittel
Eintrittswahrscheinlichkeit	Mittel
Folgewirkung	2.
Verantwortlichkeit	Kögl, Perl, jew. Rolle

6. Projekt-internes Wissen geht verloren

Verlust an Projekt-internem Wissen durch verschiedene Ursachen (zB Ausscheiden eines Projektmitglieds, Server-Ausfall, etc)

Gegenmaßnahmen:

- Dokumentation aller relevanten Informationen in Trac
- eigenständige Backups von Trac

Art	Allgemeines Risiko
Typisierung	Management
Priorität	Hoch
Eintrittswahrscheinlichkeit	Niedrig
Folgewirkung	2.
Verantwortlichkeit	Kögl, Perl, jew. Rolle

7. Unbrauchbare Daten-Basis

Sollte sich während des Testens herausstellen, dass es für Zutaten zu viele Schreibweisen gibt (zB 15 unterschiedliche Öl-Sorten), könnte der Datenbestand leicht unbrauchbar werden.

Gegenmaßnahmen:

- Den Benutzer bei ähnlichen Texten auffordern, die Zutaten zu gruppieren
- automatisch sehr ähnliche Schreibweisen als gleich betrachten (Tomate und Tomate'"n"')

Art	Projekt-spez. Risiko
Typisierung	Entwicklung, Umfeld
Priorität	Mittel
Eintrittswahrscheinlichkeit	Mittel
Verantwortlichkeit	Kögl, Perl, Wieser, Zoller

Komponentendiagramm

Das Komponentendiagramm beschreibt die Verwendung von Interfaces.

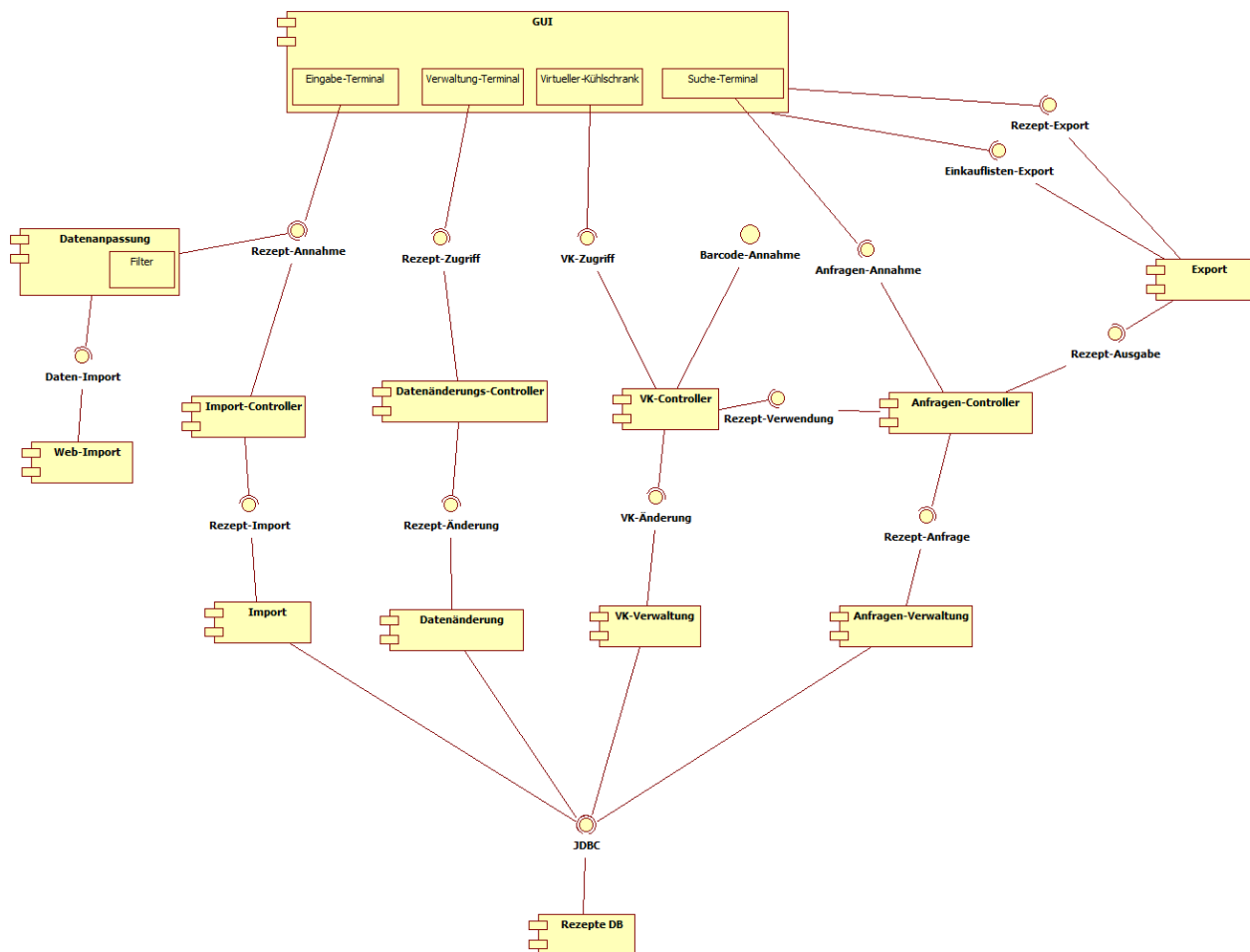


Abbildung 12: Komponentendiagramm

Unsere Software wird in elf Komponenten zerlegt. Direkten Zugriff auf das System erhält der User durch die GUI-Komponente. Hierbei sind die verschiedenen Terminals zu unterscheiden. Es gibt die Unterscheidungen:

- Eingabe-Terminal für neue Rezepte, mit dem über das Interface Rezept-Annahme auf die Komponente Import-Controller zugreifen kann.
- Suche-Terminal für die Suche nach Rezepten, so kann man die Suchanfrage über das Interface Anfragen-Annahme auf die Komponente Anfragen-Controller weitergeben und dann auf die Rezeptvorschläge warten. Diese werden danach am Terminal ausgegeben.
- Verwaltung-Terminal für Korrektur, hierbei hat man durch das Interface Rezept-Zugriff die Möglichkeit mit der Komponente Datenänderungs-Controller zu arbeiten.
- Virtueller Kühlschrank, hier hat man über das Interface VK-Zugriff eine direkte Verbindung zum VK-Controller.

Dem MVC-Konzept entsprechend verbinden Controller Klassen für Geschäftslogik und Anzeige miteinander.

In der Komponente VK-Verwaltung werden die Änderungen im Virtuellen Kühlschrank über JDBC in die Datenbank eingetragen. In der Komponente Import werden die bereits formatierten Rezepte über JDBC korrekt in die Rezepte Datenbank eingetragen. Durch die Komponente Anfragen-Verwaltung werden ankommende Such-Anfragen über JDBC an die Rezeptedatenbank gestellt und auf dem selben Weg kommen die Ergebnisse zurück. Diese müssen im Anfragen-Controller auf die Korrektheit kontrolliert werden. Die Komponente Datenbank-Korrektur ist für die Korrektur der Rezepte und anderer Tabellen bereitgestellt. Dies erfolgt über JDBC auf die Rezeptedatenbank.

Ein weiterer Weg Rezepte in die Datenbank zu bringen ist die Komponente Web-Import. Diese stellt das Interface Daten-Import zur Verfügung. Auf diesem kommen die Daten zur Komponente Datenanpassung, wo sie ins entsprechende Rezepteformat gebracht werden. Nachher werden sie über das Interface Rezept-Import an die Komponente Import weitergeleitet.

Um die Rezepte aus dem System weiterzuverarbeiten ist die Komponente Export vorhanden. Diese kommuniziert über das Interface Rezept-Export mit der Komponente GUI und über das Interface Rezept-Ausgabe mit dem Anfragen Controller. Außerdem ist diese Komponente in der Lage Einkaufslisten zu exportieren. Dies erfolgt über das Interface Einkaufslisten-Export.

Lieferkomponenten

Beim Abschluss des Projektes wird dem Kunden die erstellte Software sowie die dazugehörige Dokumentation übergeben.

Im Folgenden werden die zu übergebenden Komponenten beschrieben und eine Abgrenzung des Lieferumfangs durchgeführt.

Software

Dem Kunden wird die im Rahmen des Projekts erstellte Software übergeben, die die Spezifikationen der Anforderungsdokumente komplett abdeckt.

Die Software wird in Form einer oder mehrerer JAR-Dateien übergeben, über die der Kunde das Programm starten kann.

Die Tests, die im Zuge des Test Driven Development während der Entwicklung des Projekts erzeugt werden, werden in einer ZIP-Datei mit Sourcecode zur Verfügung gestellt.

Eventuell für das Ausführen der Software nötige Voraussetzungen sind im Benutzerhandbuch beschrieben, das ebenfalls übergeben wird.

Design-Dokumente

Dem Kunden werden Designdokumente übergeben, die während der Projektlaufzeit für die Umsetzung der Software erstellt wurden.

Diese enthalten Informationen über den internen Aufbau der Software sowie Design-Entscheidungen.

Die Design-Komponenten werden im HTML-Format geliefert.

Benutzerhandbuch

Das Benutzerhandbuch erklärt alle Funktionen der Software in für die Zielgruppe verständlicher Form.

Weiters sind im Benutzerhandbuch alle notwendigen Schritte zur Inbetriebnahme der Software beschrieben.

Das Benutzerhandbuch wird im PDF-Format geliefert.

Anforderungsdokumente

Die zu Beginn des Projektes Erarbeiteten Anforderungen an die Software sind in den Anforderungsdokumenten enthalten.

Die Anforderungsdokumente werden im HTML-Format geliefert.

Abgrenzung des Lieferumfangs

Explizit nicht im Lieferumfang enthalten sind

- Externe Software, die während der Entwicklung verwendet wurden
- Dokumente für die interne Organisation, wie Stundenaufzeichnungen, Planung, archivierte Dokumentation, etc.