

**Philosophische** Fakultät III

Sprach- , Literatur- und Kulturwissenschaften

Institut für Information und Medien, Sprache und Kultur (I:IMSK)  
Lehrstuhl für Medieninformatik

Seminar KI für Serious Games

Modul: INF-M07.2

Leitung: Professor Bernd Ludwig

Abgabedatum: 31.03.2018

**Automatische Nährstoffberechnung für den Einsatz in einem Food-Recommendation-System**

David Halbhuber 1744590

Jakob Fehle 1770881

[1 Einführung 3](#_Toc509965113)

[2 Konzeption Food-Recommandation-System 5](#_Toc509965114)

[3 Abgrenzung der Projektziele 6](#_Toc509965115)

[3.1 In scope 7](#_Toc509965116)

[3.2 Out of scope 7](#_Toc509965117)

[3.3 Konzeption Programmablauf (Einsatzszenario) 8](#_Toc509965118)

[4 Projektumsetzung 9](#_Toc509965119)

[4.1 Schnittstellenentwicklung zu Nährwertbestimmung/ Bewertung der kochbar-Datenbank 9](#_Toc509965120)

[4.1.1 Export der kochbar-Datenbank als json 10](#_Toc509965121)

[4.1.2 Anpassung an Hr.Ullmanns Software 11](#_Toc509965122)

[4.1.3 Berechnung Nährwerte und Reimport in kochbar-Datenbank 12](#_Toc509965123)

[4.2 Entwicklung Frontend zur Rezeptsuche 14](#_Toc509965124)

[4.3 Entwicklung User Interface 14](#_Toc509965125)

[5 Zusammenfassung und Ausblick 14](#_Toc509965126)

[5.1 Optimierung der Datenbanksuche 15](#_Toc509965127)

[5.2 Optimierung des Datenbankdesigns 15](#_Toc509965128)

[5.3 Optimierung des Umgangs mit Manuel Ullmanns Software 15](#_Toc509965129)

[5.4 Optimierung der Referenzwert-Tabelle 16](#_Toc509965130)

[6 Acknowledgment 17](#_Toc509965131)

[Literaturverzeichnis 18](#_Toc509965132)

# Einführung

*„Overweight and obesity impact on a child’s quality of life, as they face a wide range of barriers, including physical, psychological and health consequences. We know that obesity can impact on educational attainment too and this, combined with the likelihood that they will remain obese into adulthood, poses major health and economic consequences for them, their families and society as a whole.“*

Dr. Sanina Nishtar, Co-Chair of ECHO[[1]](#footnote-1)

Bereits im Jahre 2015 warnte die Weltgesundheitsorganisation (WHO[[2]](#footnote-2)) in ihrem Abschlussbericht[[3]](#footnote-3) zur ECHO[[4]](#footnote-4)-Kommission vor immer weiter steigenden Zahlen übergewichtiger Kinder. So berichtet die Organisation, dass der Anteil an übergewichtigen Kleinkindern unter fünf Jahren in den Jahren von 1990 bis 2014 um 1,3 Prozent gestiegen ist. In absoluten Zahlen bedeutet das ein weltweiter Anstieg von circa 10 Millionen übergewichtigen Kleinkindern. Zudem konnte die WHO zeigen, dass der Anteil Übergewichtiger mit fortschreitendem Alter ebenfalls zunimmt. Daraus wird in dem Abschlussbericht gefolgert, dass Kinder, die bereits in frühen Jahren übergewichtig sind auch im Erwachsenenalter übergewichtig sein werden. Die Weltgesundheitsorganisation beschreibt in ihrem Bericht einen sechsdimensionalen Aktionsplan um Fettleibigkeit bei Kindern zu bekämpfen.

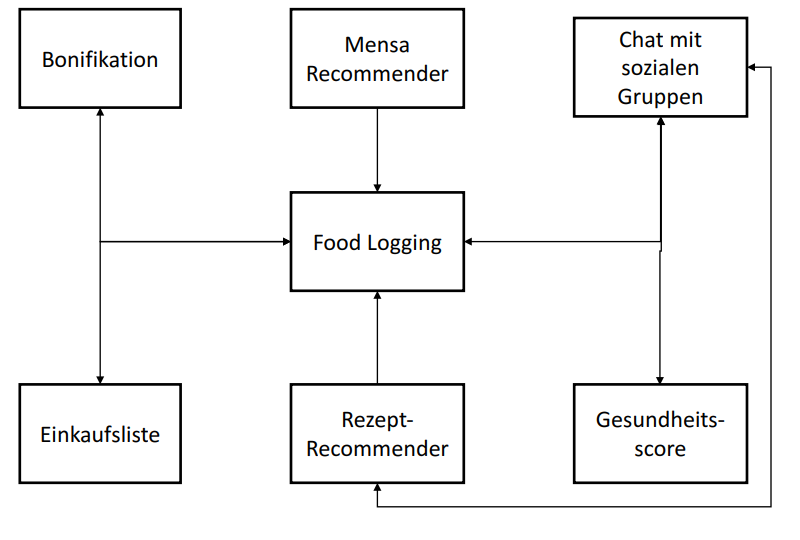
1. Promote intake of healthy foods
2. Promote physical activity
3. Preconception and pregnancy care
4. Early childhood diet and pyhsical activities
5. Health, nutrition and physical activity for schoolage children
6. Weight management

An dieser Stelle knüpft diese Projektarbeit an und versucht eine Schnittstelle bereitzustellen, um Teilaspekte dieses Aktionsplans umzusetzen. So zielt das Seminar „*KI for serious games*“, in dessen Rahmen auch diese Seminararbeit entstanden ist, darauf ab, Punkt 1 des Aktionsplans, also das Promoten von gesundem Essen, mithilfe einer Anwendung zu realisieren. Dabei werden bekannte Aspekte der Gamification und Gratifikation, wie etwa das Sammeln von Punkten dazu verwendet einen Anreiz zu schaffen, um den Benutzer im Rahmen der Anwendung zu motivieren, sich gesund zu ernähren. Die entstandene Applikation basiert dabei teilweise auf einer früheren Projektarbeit von Herrn Manuel Uhlmann[[5]](#footnote-5), welche die Berechnung von Nährstoffen anhand von Rezeptangaben ermöglicht. Die Software von Herrn Uhlmann verwendet dazu Rezepteinträge der Webseite [www.chefkoch.de](http://www.chefkoch.de) im HTML-Format. Nach einem erfolgreichen Softwareaufruf liefert die Applikation einen Vektor mit allen wichtigen Nährstoffen und Inhaltsangaben zurück. Aufbauend auf dieser Rückgabe vergleicht die, in dieser Seminararbeit entstandene Software, den Vektor mit den offiziellen Empfehlungen[[6]](#footnote-6) der deutschen Gesellschaft für Ernährung für den täglichen Konsum von Nährstoffen. Einem Benutzer soll so die Möglichkeit gegeben werden seinen täglichen Nährstoffbedarf zu überwachen. Dabei sollen dem Nutzer vergangene Mahlzeiten angezeigt werden, um so eine Art Mahlzeiten-Tagebuch zu führen. Die gesamte Anwendung ist dabei in ein Food-Recommendation-System eingebettet, welches nicht innerhalb dieser Arbeit entwickelt wird, aber im Rahmen des gesamten Seminars entstehen soll. Basierend auf den eigenen Ernährungszielen eines Users schlägt das System verschieden Mahlzeiten aus einer Datenbank vor. Der User kann dabei entweder eines der vorgeschlagenen Gerichte wählen oder sich über eine Suchfunktion eine andere Speise aussuchen. Nach der Wahl der Speise wird dies im User-Log gespeichert und die Anzeige für den täglichen Nährstoffbedarf entsprechend angepasst. Für die in dieser Arbeit entwickelten Software wird angenommen, dass das Food-Recommendation-System bereits etabliert ist und bei Benutzerinteraktion einen API-Call an die hier entworfene Schnittstelle ausführt. Die entstandene Projektarbeit ist dabei wie folgend gegliedert: Das nächste Kapitel skizziert das gesamte „Ökosystem“ des Food-Recommendation-System in welches unsere Schnittstelle eingebettet ist. Im darauf folgenden Kapitel wird noch einmal detailliert abgegrenzt welche Komponenten Bestandteil dieser Arbeit sind. Im vierten Kapitel werden kurz erste Paper-Sketches gezeigt und mögliche User-Interaktion konzipiert. Im fünften Kapitel dieser Arbeit wird die eigentliche Entwicklung der Schnittstelle zur automatischen Nährstoffberechnung aufgezeigt. Schlussendlich folgen eine kurze Zusammenfassung und ein Ausblick für zukünftige Arbeiten.

# Konzeption Food-Recommendation-System

Dieses Kapitel soll die komplette Softwareumgebung der Schnittstelle zur automatischen Nährwertberechnung skizzieren und konzeptionell darstellen. Dabei sind alle erwähnten Komponenten und Funktionen der Umgebung theoretische Konstrukte und noch nicht in dem erwähnten Umfang verfügbar. Oberstes Ziel des Food-Recommendation-Systems ist es, einen Benutzer zu einer gesunden, bzw. gesünderen Ernährung anzuhalten. In einem ersten Schritt soll dem Benutzer die Möglichkeit geboten werden sein persönliches Ernährungsziel zu wählen, wobei zwischen verschiedenen Zielen, wie etwa Gewichtsabnahme oder Muskelaufbau, unterschieden werden kann. Basierend auf diesem Ernährungsziel entwickelt das Food-Recommendation-System erste Vorschläge für Rezepte oder Mahlzeiten. Dabei kann der Benutzer seine persönlichen Präferenzen für Mahlzeiten oder Rezepte in einer Art „FoodTinder“ durch „Swipen“ auswählen. So können zum Beispiel Mahlzeiten und/oder Rezepte mit Pilzen ausgeschlossen oder Mahlzeiten mit Spinat bevorzugt werden. Nach dem Definieren der eigenen Vorlieben für Speisen, bzw. dessen Bestandteile erarbeitet das Recommendation-System Vorschläge für Mahlzeiten. Der Benutzer kann nun eine oder mehrere der angezeigten Mahlzeiten wählen oder kann sich neue Vorschläge generieren lassen. Durch die Verwendung der Anwendung kann der Benutzer Punkte sammeln, z.B. wenn Gerichte gewählten werden die zum eigenen Ernährungsziel passen.

Desweiteren soll die Applikation auch eine soziale Komponente implementieren. Dem Benutzer soll es ermöglicht werden sich mit Bekannten und Freunden in der App auszutauschen. So soll es möglich sein, eigene Mahlzeiten innerhalb eines Gruppenchats zu teilen. Die Teilnehmer können die geteilten Mahlzeiten dann bewerten und so ebenfalls Punkte verdienen. Die gesammelten Punkte der Benutzer können schließlich zum Einkaufen genutzt werden, wobei ungesunde Lebensmittel dabei mehr Punkte verbrauchen, als gesunde Alternativen.



*Abbildung 1: Architektur des Food-Recommendation-Systems*

In Abbildung 1 sind eben genannte Komponenten zu sehen und zeigt wie diese miteinander interagieren sollen. Der Fokus dieser Arbeit liegt dabei auf Berechnung und Visualisierung des täglichen Nährstoffbedarfs eines Benutzers.

# Abgrenzung der Projektziele

Kapitel 2 hat gezeigt, dass es sich bei dem entstehenden Food-Recommendation-System um ein komplexes System aus vielen verschiedenen Komponenten handelt, die schlussendlich in einer finalen Version zusammengeführt werden sollen. Durch diesen hohen Grad an Komplexität entsteht die Notwendigkeit klar zu definieren welche Funktionalitäten im Rahmen dieser Projektarbeit entwickelt werden sollen und welche Funktionalitäten nicht Bestandteil dieser Arbeit sind und als gegeben angenommen werden müssen.

## In scope

Als „in scope“, sollen folgende Funktionalitäten verstanden werden, die im Rahmen dieser Arbeit konzeptioniert und entwickelt wurden. Dabei reduziert sich der Funktionsumfang auf vier große Teilgebiete:

* Entwicklung einer Schnittstelle zwischen dem Benutzer und der entwickelten Software von Manuel Ullmann um Rezepte und Mahlzeiten in Echtzeit analysieren zu können.
* Bewertung des kochbar-Datensatzes auf Nährwerte, welcher im Rahmen des Seminars an der Universität Regensburg zur Verfügung gestellt wurde.
* Entwicklung eines Frontend‘s zum Durchsuchen des kochbar-Datensatzes.
* Die Visualisierung und benutzerorientierte Speicherung des täglichen Nährwertbedarfs, sowie die Speicherung von bereits verzehrten Speisen.

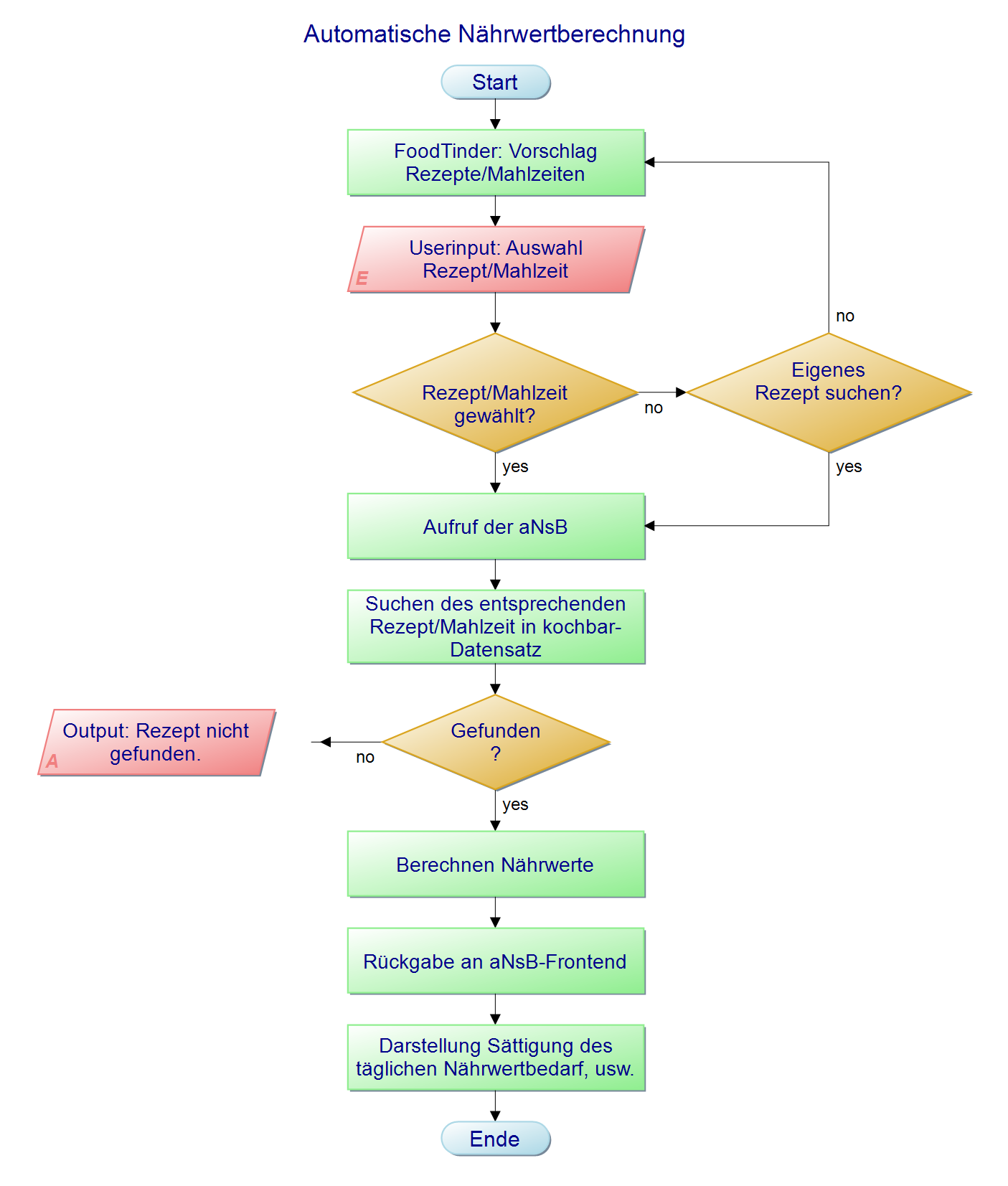
## Out of scope

Als „out of scope“ werden in dieser Arbeit alle Funktionalitäten bezeichnet, die nicht im Umfang dieser Arbeit enthalten sind. Diese Funktionalitäten müssen, um die Entwicklung einer Schnittstelle zur automatischen Nährstoffberechnung zu ermögliche, als gegeben und funktional angesehen werden. Dabei werden nur Komponenten simuliert die für die Anwendung unbedingt notwendig sind, dies setzt sich zusammen aus:

* Eine funktionierende „FoodTinder“ Funktion die, die Schnittstelle zur Nährwerberechnung aufruft.
* Eine funktionierende „FoodTinder“ Funktion die, für den Fall das der Benutzer seine Speise selbst suchen möchte, auf das entwickelte Frontend verweist.
* Eine funktionierende Benutzerverwaltung die bei Schnittstellenaufruf eine eindeutige Benutzerdaten (ID, Alter, Geschlecht) liefert. Die Benutzeridentifikation wird in dieser Arbeit durch einen „Dummy“ Eintrag simuliert.

## Konzeption Programmablauf (Einsatzszenario)

Der in *Abbildung 2* zu sehende Programmablaufplan diente als Grundlage für weiterführende Konzepte und soll schematisch Darstellen wie der Benutzer mit Software interagiert.



*Abbildung 2: Programmablaufplan[[7]](#footnote-7) „Automatische Nährwertberechnung“*

Der Programmablaufplan startet mit der Präsentation einer Rezeptauswahl des „FoodTinder“-Algorithmus. Der Benutzer hat nun die Möglichkeit aus einer Auswahl von Rezepten/Gerichten jenes zu wählen, welches am ansprechendsten erscheint, bzw. das Gericht, das am besten zu dem individuellen Ernährungsziel passt. Sollte dem Anwender keines der angezeigten Gerichten zusagen, bietet das System die Möglichkeit entweder neue Empfehlungen auszusprechen, oder selbst nach einem Rezept in der Datenbank zu suchen. Im ersten Fall wiederholt sich dieser Vorgang solang bis dem Benutzer ein passendes Gericht vorgeschlagen wurde - dieses Rezept wird dann dem Frontend übergeben. Im zweiten Fall wird der Benutzer auf das Frontend zur Rezeptsuche weitergeleitet - nun kann der Benutzer ein eigenes Gericht auswählen oder erstellen. In beiden Fällen wird für das final gewählte Gericht eine Berechnung gestartet, um dessen Nährwerte zu erhalten. Zuletzt werden dem Benutzer diese Nährwerte zusammen mit ausgewähltem Gericht im Frontend angezeigt.

# Projektumsetzung

Die folgenden Kapitel dokumentieren die einzelnen Unterbereiche des gesamten Projekts.

## Schnittstellenentwicklung zu Nährwertbestimmung/ Bewertung der kochbar-Datenbank

Der erste praktische Teil dieser Arbeit bestand in der Entwicklung einer Schnittstelle zur Kommunikation mit der bereitgestellten Software von Hr. Manuel Ullmann, sowie die Berechnung der Nährwerte aller Rezepte in der kochbar-Datenbank.

In der uns zugänglichen Version der Software akzeptiert diese als Aufrufparameter nur sogenannte „.raw“ Dateien. Diese Dateien beinhalten eine html-Kopie des Seitenquelltextes von Rezepten der Seite www.chefkoch.de„. Die Vorgehensweise, wie Herr Ullmanns Software Daten erfasst und verarbeitet stellte sich für dieses Projekt als problematisch heraus. Die Software durchsucht die „.raw“-Dateien nach bestimmten HTML Tags und wertet diese aus, um Inhaltsangaben, Mengenangabe, Rezepttitel und andere Daten über gewählte Gerichte zu erhalten. Der uns vorgegebene Datensatz, die kochbar-Datenbank, beinhaltet diese HTML Tags nicht. Ohne weitere Änderung wäre die Auswertung der kochbar-Datenbank nicht möglich gewesen, da Rezepte in der Datenbank als SQL-Eintrag vorliegen. Um die Bewertung der kochbar-Datenbank trotzdem mit der von Herrn Ullmann bereitgestellten Software zu ermöglichen, wurden an dieser Anpassungen vorgenommen ,sodass diese auch „JSON“ Dateien als Eingabeparameter annimmt und verarbeiten kann. Um schlussendlich die Berechnung der Nährwerte der Einträge der kochbar-Datenbank zu realisieren, folgte dieser Teil der Arbeit folgendem Schema:

1. Export der kochbar-Datenbank als JSON Dateien
2. Anpassen der Software von Herrn Ullmann, um Berechnungen von Rezepten im JSON Format zu ermöglichen
3. Berechnung der Nährwerte der kochbar-JSON Dateien
4. Reimport in kochbar-Datenbank

### Export der kochbar-Datenbank als JSON

Zum Export der kochbar-Datenbank wurde die Software MySQL Workbench[[8]](#footnote-8) verwendet. Der bereitgestellte Datenbankexport wurde dabei zur Bearbeitung lokal gehostet. Zur Berechnung der Nährstoffe eins Gerichts benötig die Software von Herr Ullmann lediglich dessen Zutaten. Diese, zusammen mit einem eindeutigen Identifier, werden in der kochbar-Datenbank in der Tabelle **kochbar.kochbar\_recipes** dargestellt. Der ehemalige Link des Rezeptes[[9]](#footnote-9) wird dabei als zuvor genannter Identifier genutzt. Identifikation musste ebenfalls mit exportiert werden, um so in späteren Schritten eine Zuordnung der berechnetet Nährwerte zu den ursprünglichen Rezepten zu ermöglichen. Schlussendlich beinhaltet der Export also die href[[10]](#footnote-10) Adresse des Rezepts und dessen Zutaten. Da die Tabelle in der Summe 309360 Rezepte beinhaltet und damit die Dateigröße der entstehenden JSON-Datei zu groß geworden wäre, wurden jeweils 30000 Rezepte auf einmal exportiert.



*Abbildung 3: Query zum Export der Rezepte*

Die Limitierung auf mehrere, kleine JSON Dateien hat auch den Vorteil, dass im späteren Verlauf, bei der Vewendung von Herrn Ullmanns Software, auf Techniken der parallelen Datenverarbeitung zurück gegriffen werden kann. Parallelisierung hat den offensichtlichen Vorteil, dass mit ihrer Hilfe die nötige Zeit der Datenverarbeitung drastisch reduziert werden kann. Vor allem im Umfeld der Datenbankentwicklung ist dies gängige Praxis.

Die in diesem Vorgang entstanden JSON Dateien können dem, für diese Arbeit verwendeten GitHub Repository[[11]](#footnote-11), entnommen werden.

### Anpassungen an Hr.Ullmanns Software

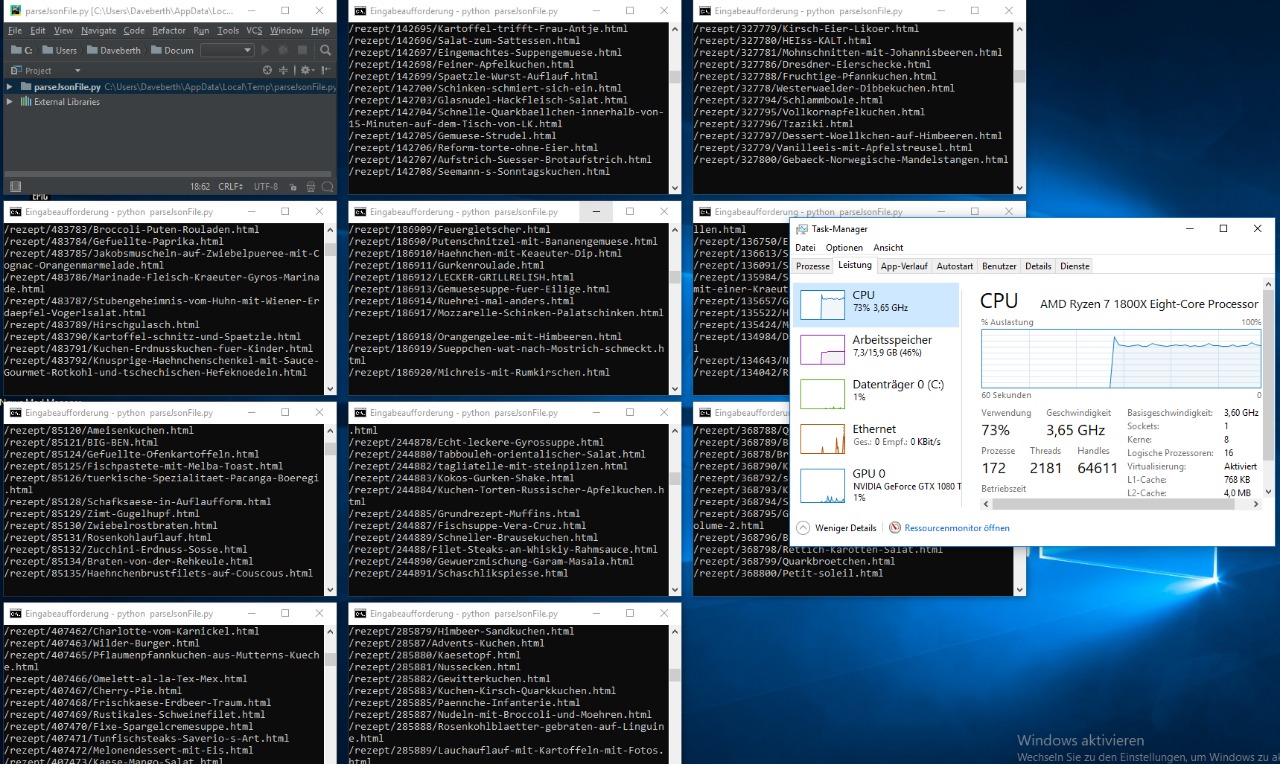
Wie bereits in der Einführung des Kapitels 4 erwähnt, mussten an der Software einige Änderungen vorgenommen werden, damit diese mit das JSON-Dateiformat verarbeiten kann. Der grundsätzliche Programmaufruf für die Berechnung der Nährwerte[[12]](#footnote-12) erfolgte über den Aufruf der die **readprep.py**-Datei mit einer Rezeptdatei. Die **readprep.py**-Dateiimportierte bei ihrem Aufruf **findnutr.py**-Datei. Da die **readprep.py**-Datei für das Projekt keinen Nutzen hat, wurde der Einstiegspunkt abgeändert. Im Ursprungsprogramm war die **readprep.py** u.a.dafür zuständig aus der Rezeptdatei die Rezeptbeschreibung und die Anleitung zum Kochen zu parsen. Für die Berechnung der Nährwerte sind diese Informationen allerdings nicht von Bedeutung.

Im Anschluss musste die Software abgeändert werden, um das Verarbeiten von JSON-Strings zu ermöglichen, wobei die Rezeptklasse in **findnutr.py** angepasst wurde. Die Klasse wird nun mit einen JSON Objekt instanziiert, aus welchem diese die entsprechenden Informationen zur Berechnung der Nährwerte erhält. Dafür wurde die Funktion „*\_\_readJson*“ zum Einlesen einer JSON Datei, die Funktion „*\_\_parseJson*“ zum Parsen der eingelesenen JSON Datei, sowie die Funktion „*returnJson*“ zum Export des Ergebinsses als JSON Datei,[[13]](#footnote-13) implementiert. Alternativ können die errechneten Nährwerte auch als Array abgefragt werden. Die entstandene **findnutr.py** kann ebenfalls im GitHub-Repository gefunden werden.

### Berechnung Nährwerte und Reimport in kochbar-Datenbank

Der letzte Schritt der Schnittstellenentwicklung bestand darin, mit der final angepassten, Software von Herrn Ullmann für die im JSON-Format vorliegenden Rezepte die Nährwerte zu berechnen. Dieser Vorgang steht auch exemplarisch dafür wie zukünftig andere Entwickler und Benutzer mit der in dieser Arbeit entwickelten Schnittstelle interagieren können. Prinzipiell stützt sich die Bewertung auf ein weiteres Python Skript, **writeToDb.py**. Dieses Skript ist dafür zuständig, die Einträge, die in dem Array der **findnutr.py** gesichert wurden in eine neue Tabelle der Datenbank zu speichern. Die in diesem Vorgang nötigen SQL-Befehle wurden dabei durch Wildcards maskiert um der Gefahr durch SQL-Injektion entgegen zu wirken. Desweiteren überprüft die **writeToDb.py**-Datei**,** ob die passende Tabelle auf der Datenbank bereits existiert und legt diese im Fall an. Die eigentliche Berechnung wird nun durch den Aufruf der **findnutr.py**-Dateigestartet. Dabei erwartet das Python Skript nun ein oder mehrere Rezepte im JSON-Dateiformat. Die Nährwerte der entsprechenden Gerichte werden berechnet und via **writeToDb.py**-Dateiin die neue Tabelle der Datenbank geschrieben. Zur Berechnung der Nährwerte des kochbar-Datensatzes wurde die Funktion 11-mal gestartet, mit jeweils 30000 unterschiedlichen Rezepten. Die Parallelisierung der Berechnung und dem Schreiben in die Datenbank kann auf zwei Wegen erreicht werden. Entweder wird die **findnutr.py** manuell so oft wie nötig gestartet oder das im GitHub Repository beiliegende Python Skript **scriptHandler.py** wird genutzt. Das **scriptHandler.py**-Skript startet ein beliebiges drittes Skript, beliebig oft, mit beliebigen Parametern.

Abbildung 4 zeigt den Reimport Vorgang der Rezepte mit den entsprechenden Nährwerten. Dank Parallelisierung konnten innerhalb von 5 Stunden 309360 Rezepte auf ihre Nährwerte untersucht und die Ergebnisse in eine Datenbank geschrieben werden. Auf diese Weiße können auch neue Rezept zur bestehenden Datenbank hinzugefügt werden, wobei die Anwender lediglich wie oben beschrieben das **findnutr.py** Skript mit einer passenden JSON Datei aufrufen müssen. Die Formatierung der json Datei kann dabei der im GitHub-Repository zu findenden **testparse.json** entnommen werden.



*Abbildung 4: Reimport der Rezept-Nährwert-Kombination*

Zuletzt muss für den Reimport die eben angelegte Tabelle noch als SQL-Dump exportiert werden, um später auf dem Universitätsserver integriert werden zu können

## Entwicklung Frontend zur Rezeptsuche

## Entwicklung User Interface

# Zusammenfassung und Ausblick

Dieses Kapitel soll die vorliegende Arbeit noch einmal zusammenfassen und zusätzlich einen Ausblick auf mögliche anknüpfende Arbeiten geben.

Es konnte gezeigt werden, wie mit Hilfe der von Herrn Ullmann entwickelter Software ca. 300.000 Rezepte auf ihre Nährwerte hin untersucht wurden. Des Weiteren wurden die errechneten Nährwerte in einer neuen Tabelle der kochbar-Datenbank gesichert und ermöglichen so anderen Seminarteilnehmern oder künftigen Entwicklern einen direkten und einfachen Zugriff auf die entsprechenden Werte, ohne diese selbstständig berechnen zu müssen. Zudem konnte in dieser Arbeit demonstriert werden, wie Anwender zukünftig auch Rezepte bewerten können, ohne dass diese zwingend dem Repertoire der Internetseite [www.chefkoch.de](http://www.chefkoch.de) entspringen. Die Rezepte oder Gerichte müssen dazu lediglich in einer passenden JSON-Datei vorliegen. Im zweiten und dritten praktischen Teil dieser Arbeit wurde ein Interface zur Kommunikation mit der kochbar-Datenbank entwickelt und präsentiert. Dabei hat der Anwender die Möglichkeit Rezepte aus der Datenbank zu suchen und diese in seinem Dashboard, also in seinem persönlichen Bereich, abzulegen. Das Dashboard des Users zeigt dabei seinen täglichen Bedarf an verschiedenen Nährstoffen an und trifft auch eine Aussage darüber, wieviel der User davon bereits durch ausgewählte Mahlzeiten aufgenommen hat. Die in Kapitel 3.1 vorgestellten Projektziele sind damit in vollem Umfang erfüllt worden.

Leider haben sich während der Projektdurchführung einige Schwachstellen und konzeptionelle Fehler mir der eingesetzten Methodik aber auch mit den verwendeten Programmen und Skripten, gezeigt. Diese sollen im Folgenden kurz aufgezeigt um damit eventuelle Ansatzpunkte für zukünftige Projekte und Arbeiten zu liefern.

## Optimierung der Datenbanksuche

In der aktuellen Version des in dieser Arbeit entwickelten UIs (User Interface) gibt es keine Möglichkeit die Datenbank nach Rezeptbestandteilen bzw. nach bestimmten Zutaten zu durchsuchen. Während des Suchvorgangs wird lediglich die Spalte des Rezepttitels, also der eindeutige Identifier, des Gerichts beziehungsweiße der Mahlzeit durchsucht. Dies hat den Hintergrund, dass sich das Durchsuchen der Zutaten-Spalte von ca. 300000 Rezepten als extrem ineffizient erwiesen hat. Zu Gunsten der Responsivität des Interfaces und der erfahrenen User Experience implementiert diese Arbeit lediglich eine Suche in der Titelspalte. Es wäre aber durchaus wünschenswert, dass User auch nach bestimmten Zutaten suchen oder diese, falls gewünscht, auch ausschließen können, ohne, dass diese zwingend im Namen des Gerichts vorkommen. Daher wäre die Entwicklung einer performanten Volltextsuche der Inhaltsstoffspalte für die kochbar-Datenbank eine mögliche Thematik für zukünftige Arbeiten.

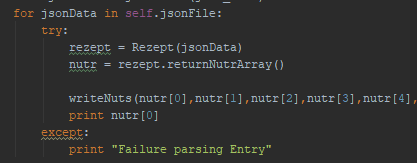
## Optimierung des Datenbankdesigns

Die vorliegende, von uns entwickelte Tabelle, also die der Zuordnung von Rezepttiteln und Nährwerten, verzichtet auf die Verwendung von traditionellen Datenbankattributen. So wird in der Tabelle keine Referenz auf die eigentlich ursprüngliche Rezepttabelle angelegt. Üblicherweise wird hier der Titel der Rezepte in der Tabelle als FOREIGN KEY gesetzt, da jedoch dies die Modularität der entstandenen Tabelle gemindert hätte und der mögliche Einsatzzweck der Tabelle über den Rahmen dieses Projektes hinausgeht, wurde sich dagegen entschieden einen FOREIGN KEY auf den PRIMARY KEY der ursprünglichen Rezepttabelle zu setzen. Ähnlich verhält es sich auch mit den gewählten Datentypen der Spalten der neuen Tabelle. Diese sind möglichst variabel und flexibel gehalten um Anpassungen in der Tabelle auch für anderen Entwicklern so möglich wie einfach zu gestalten. Für spätere Arbeiten wäre es sicherlich erstrebenswert das Datenbankdesign entsprechend zu optimieren.

## Optimierung des Umgangs mit Manuel Ullmanns Software

Es zeigte sich, dass die Software von Herrn Ullmann gelegentlich Probleme damit hat die Mengenangaben in Rezepten korrekt zu interpretieren. Als Folge daraus können für das Gericht/Rezept keine Nährwerte bestimmt werden. In dieser Arbeit konnte der Ursprung des Problems nicht klar ausgemacht werden. So scheint die Software Probleme mit dem Erkennen mancher Mengenangaben zu haben.

Schlussendlich konnte, durch den Einsatz eines Eventhandlers in der **parseJsonFile.py**, die Problematik umgangen werden. Rezepte die nicht eindeutig bewertet werden konnten, wurden ausgelassen, anstatt in die resultierende Tabelle aufgenommen zu werden. Die Summe der dadurch verloren Rezepte beläuft sich bei etwa 1% des gesamten kochbar-Datensatzes – es konnten also ca. 5000 Rezepte nicht bewertet werden. Für zukünftige Arbeiten wäre die Verbesserungen der „Erkenner-Quote“ selbstverständlich interessant, vor allem um individueller auf Benutzereingaben von neuen Rezepten reagieren zu können.



*Abbildung 5: Ausschnitt des Eventhandlers*

## Optimierung der Referenzwert-Tabelle

Zum Vergleich der konsumierten Speisen und den, nach der deutschen Gesellschaft für Ernährung, empfohlen täglichen Referenzmengen für Nährstoffe verwendet diese Arbeit eine dafür angelegte Datenbanktabelle. In dieser Tabelle sind zehn verschiedene Kategorien mit jeweils unterschiedlichen Mengenangaben für den empfohlenen Nährstoffkonsum definiert. Dabei wird sowie nach dem Geschlecht des Anwenders, als auch dessen Alter (15-19 Jahre alt, 19-24 Jahre alt, 24-51 Jahre alt, 51–65 Jahre alt und 65-100 Jahre alt) unterschieden. Die deutsche Gesellschaft für Ernährung sieht allerdings noch weitere Gruppe vor, wie etwa stillende Frauen, schwangere Frauen, Kleinkinder, Säuglinge und Menschen mit bestimmten Erkrankungen. Da durch die Implementierung dieser Gruppen, in Anbetracht der zu erwartenden Nutzergruppe (Personen, deren vorrangiges Ziel eine gesündere Ernährung ist) kein Mehrwert zu erwarten gewesen wäre, wurden diese Kategorien übergangen. Für zukünftige Arbeiten könnte die Erweiterung der Tabelle, um ein Einträge für spezielle Personengruppen, aber durchaus von Interesse sein. So könnte das entwickelte Frontend zum Beispiel auch zum Diabetes-Management-Tool um konzipiert werden, um Menschen mit Zuckererkrankung beim Aufzeichnen und Überwachen ihrer Mahlzeiten zu helfen.

## Optimierung des Frontend Designs

# Acknowledgment

An dieser Stelle möchten wir uns herzlichst bei Manuell Ullmann für die Bereitstellung seiner Software bedanken, mit welcher die Umsetzung dieses Projekts erst möglich wurde.

# Literaturverzeichnis

* Deutsche Gesellschaft für Ernährung, 2017. Referenzwerte für die Nährstoffzufuhr (D-A-CH),

1. Quelle, Reuters.com: <https://www.reuters.com/article/us-health-obesity-children/number-of-obese-and-overweight-children-under-five-alarming-who-says-idUSKCN0V320W>, Abgerufen, 19.03.2018 [↑](#footnote-ref-1)
2. Homepage, Weltgesundheitsorganisation: <http://www.who.int/en/>, Abgerufen: 19.03.2018 [↑](#footnote-ref-2)
3. Abschlussbericht, „Commission on ending childhood obesity“, Quelle: <http://www.who.int/end-childhood-obesity/news/launch-final-report/en/>, Abgerufen: 19.03.2018 [↑](#footnote-ref-3)
4. Akronym für „Commission on ending childhood obesity“ [↑](#footnote-ref-4)
5. Automatisierte Berechnung von Nährwerten einer großen, heterogenen Rezeptdatenbank, Manuel Uhlmann, 2012; https://elearning.uni-regensburg.de/mod/resource/view.php?id=915461 , Abgerufen: 19.03.2018 [↑](#footnote-ref-5)
6. Homepage der deutschen Gesellschaft für Ernährung, Referenzwerte für Nährstoffzufuhr, Quelle: <https://www.dge.de/wissenschaft/referenzwerte/> ,Abgerufen: 22.03.18 [↑](#footnote-ref-6)
7. aNsB = automatische Nährwertberechnung [↑](#footnote-ref-7)
8. MySQL Workbench, Download: <https://www.mysql.com/de/products/workbench/> , Abgerufen: 10.03.2018 [↑](#footnote-ref-8)
9. [↑](#footnote-ref-9)
10. Eigentlich handelt es sich bei bei dem Term „href“ um einen HTML-Tag der dazu genutzt wird Inhalte oder Webseiten zu verlinken. Siehe dazu auch, W3Schools „href-Attribut“: <https://www.w3schools.com/tags/att_a_href.asp> ,Abgerufen: 10.03.18 [↑](#footnote-ref-10)
11. GitHub-Repository: <https://github.com/JakobFehle/KI4SG> [↑](#footnote-ref-11)
12. Die Software hat noch verschiedene andere Funktionen, in dieser Arbeite nutzten wir jedoch ausschließlich die Funktion zur Berechnung der Nährwerte. [↑](#footnote-ref-12)
13. [↑](#footnote-ref-13)