



Universität Regensburg

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 9999999

1	Einführung	3
2	Konzeption Food-Recommandation-System	5
3	Abgrenzung der Projektziele	6
3.1	In scope	7
3.2	Out of scope	7
3.3	Konzeption Programmablauf (Einsatzszenario)	8
4	Projektumsetzung	9
4.1	Schnittstellenentwicklung zu Nährwertbestimmung/ Bewertung der kochbar-Datenbank.....	9
4.1.1	Export der kochbar-Datenbank als json.....	10
4.1.2	Anpassung an Hr.Ullmanns Software	11
4.1.3	Berechnung Nährwerte und Reimport in kochbar-Datenbank.....	12
4.2	Entwicklung Frontend zur Rezeptsuche	14
4.3	Entwicklung User Interface	14
5	Zusammenfassung und Ausblick	14
5.1	Optimierung der Datenbanksuche	15
5.2	Optimierung des Datenbankdesigns.....	15
5.3	Optimierung des Umgangs mit Manuel Ullmanns Software.....	15
6	Acknowledgment	16
	Literaturverzeichnis.....	17

1 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 ECHO¹

Bereits im Jahre 2015 warnte die Weltgesundheitsorganisation (WHO²), in ihrem Abschlussbericht³ zur ECHO⁴-Kommission vor immer weiter steigenden Zahlen übergewichtiger Kinder. So berichtet die Organisation, dass der Anteil an übergewichtigen Kleinkinder 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. Außerdem 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 ein sechs-dimensionalen 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

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

² Homepage, Weltgesundheitsorganisation: <http://www.who.int/en/>, Abgerufen: 19.03.2018

³ Abschlussbericht, „Commission on ending childhood obesity“, Quelle: <http://www.who.int/end-childhood-obesity/news/launch-final-report/en/>, Abgerufen: 19.03.2018

⁴ Akronym für „Commission on ending childhood obesity“

An dieser Stelle setzt diese Projektarbeit ein und versucht eine Schnittstelle bereit zu stellen 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, mit Hilfe 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 der Anwendung zu motivieren, sich gesund zu ernähren. Die entstandene Applikation basiert dabei teilweise auf einer früheren Projektarbeit von Herrn Manuel Uhlmann⁵, welche die Berechnung von Nährstoffen anhand von Rezeptangaben ermöglicht. Die Software von Herrn Uhlmann verwendet dazu Rezepteinträge der Webseite 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 Empfehlungen⁶ der Weltgesundheitsorganisation für den täglichen Konsum von Nährstoffen. Einem Benutzer soll so die Möglichkeit gegeben werden seinen täglichen Nährstoffbedarf zu überwachen. Dabei sollen dem Nutzer vergangenen Mahlzeiten angezeigt, werden um so eine Art Mahlzeiten-Tagebuch zu führen. Die gesamte Anwendung ist dabei in ein Food-Recommandation-System eingebettet, welches nicht im Rahmen dieser Arbeit entwickelt wird, aber im Rahmen des gesamten Seminars entstehen soll. Basierend auf den eignen Ernährungszielen eines Users schlägt das System verschiedenen 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 das, dass Food-Recommandation-System bereits etabliert ist und bei Benutzer Interaktion einen API-Call an die hier entworfene Schnittstelle ausführt. Die entstandene Projektarbeit

⁵Automatisierte 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

⁶ Daily Dietary recommendations / Nutritional requirements, WHO, Quelle: <http://www.who.int/nutrition/topics/nutrecomm/en/>, Abgerufen: 19.03.18

ist dabei wie folgend gegliedert: Das anschließende Kapitel skizziert das gesamte „Ökosystem“ des Food-Recommandation-System in welches die Schnittstelle eingebettet ist. Im dritten Kapitel wird noch einmal detailliert abgegrenzt welche Komponente Bestandteil dieser Arbeit sind. Im vierten Kapitel werden kurz erste Paper Sketches gezeigt und mögliche User Interaktion konzipiert. Im fünften Teil dieser Arbeit wird die eigentliche Entwicklung der Schnittstelle zur automatischen Nährstoffberechnung aufgezeigt. Schlussendlich folgen eine kurze Zusammenfassung und ein Ausblick für zukünftige Arbeiten.

2 Konzeption Food-Recommandation-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-Recommandation-Systems ist es, einen Benutzer zu einer gesunden, beziehungsweise 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-Recommandation-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, beziehungsweise 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. So können Punkte verdient werden, wenn Gerichte gewählt werden die zum eigenen Ernährungsziel passen. Des Weiteren 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 seine eigene Mahlzeit mit einem

Gruppenchat 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, ungesunde Lebensmittel sollen dabei mehr Punkte kosten als gesunde Alternativen.

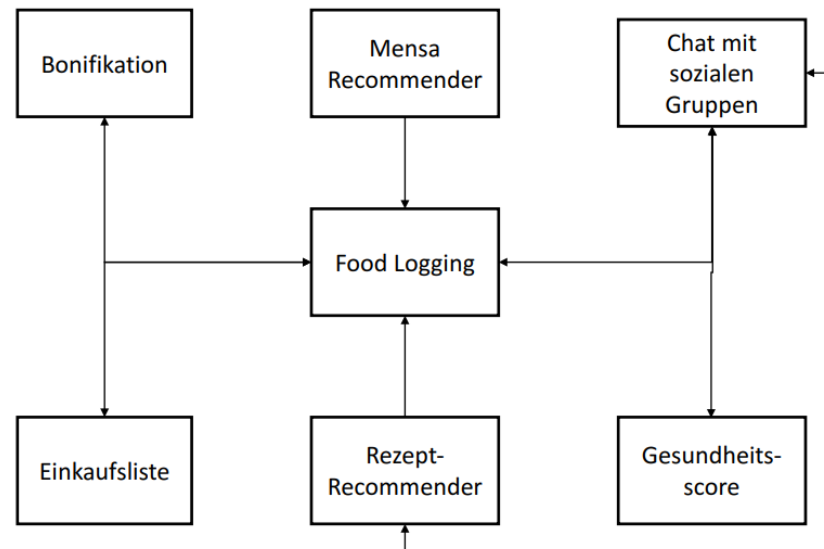


Abbildung 1: Architektur des Food-Recommendation-System

Abbildung 1 verbildlich eben genannte Komponenten und zeigt wie diese mit einander interagieren sollen. Der Fokus dieser Arbeit liegt dabei in dieser Darstellung auf den Komponenten „Food Logging“ sowie „Gesundheitsscore“ wie im folgenden Kapitel noch einmal verdeutlicht werden soll.

3 Abgrenzung der Projektziele

Wie in Kapitel 2 gezeigt wurde handelt es sich bei dem entstehenden Food-Recommendation-System um ein komplexes System aus vielen verschiedenen Komponenten 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.

3.1 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 Teilgebiet:

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

3.2 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 ermöglichen, 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 Nährwertberechnung 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 Benutzeridentifikation (ID) liefert. Die Benutzeridentifikation wird in dieser Arbeit durch einen „hart gecodeten“ Eintrag simuliert.

3.3 Konzeption Programmablauf (Einsatzszenario)

Folgender Programmablaufplan in *Abbildung 2* diene als Grundlage für weiterführende Konzepte und soll schematisch darstellen wie der Benutzer mit Software interagiert.

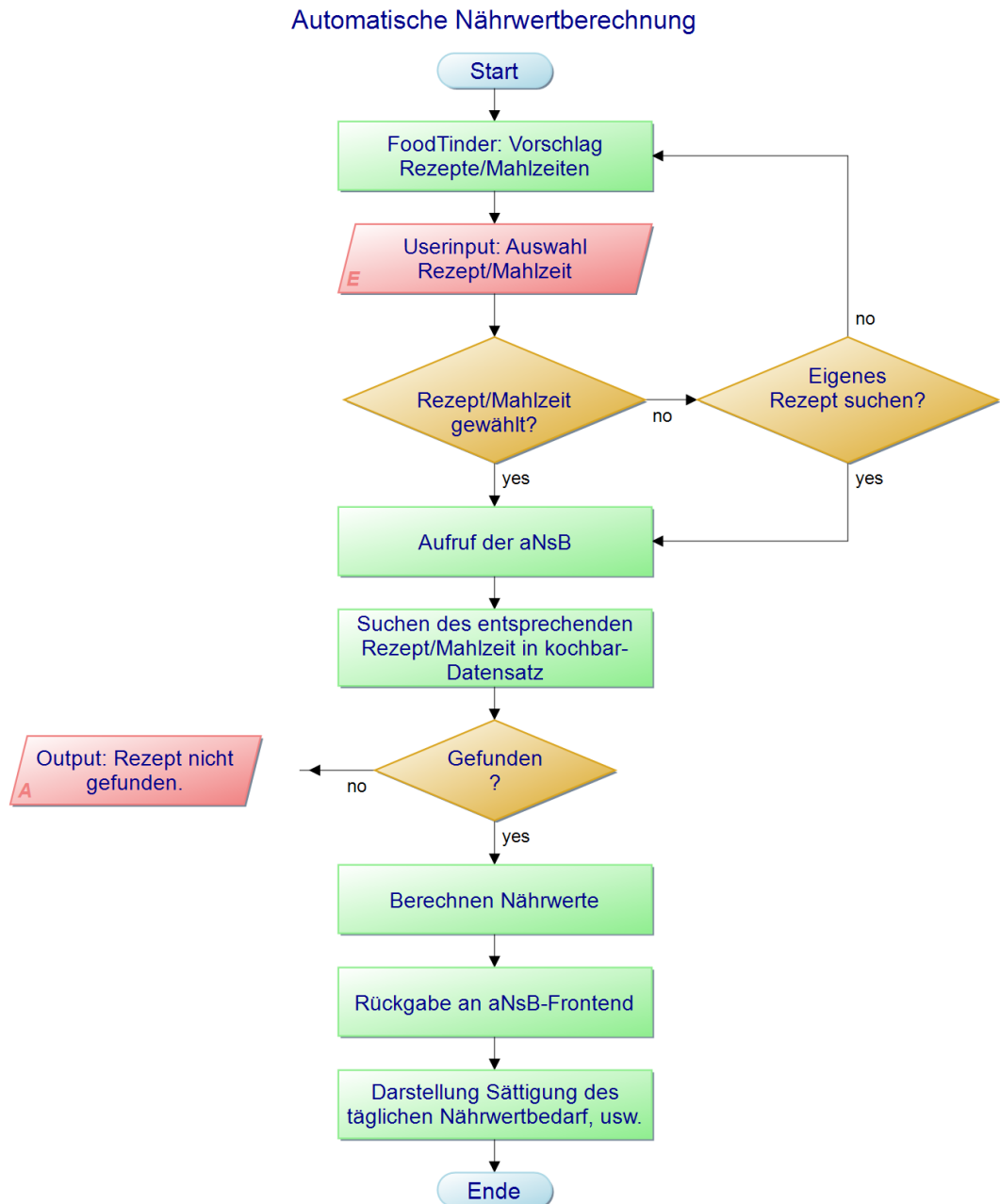


Abbildung 2: Programmablaufplan⁷ „Automatische Nährwertberechnung“

Der Programmablaufplan startet mit der Präsentation einer Rezeptausswahl des „Food-Tinder“-Algorithmus. Der Benutzer hat nun die Möglichkeit aus einem Assortiment von

⁷ aNsB = automatische Nährwertberechnung, abgekürzt um die Formatierung des PAP schlank zu halten.

Rezepten/Gerichten jenes zu wählen, das am ansprechendsten erscheint, beziehungsweise 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 solange bis dem Benutzer ein passendes Gericht vorgeschlagen wurde, dieses Rezept wird dem Frontend übergeben. Im zweiten Fall wird der Benutzer auf das Frontend zur Rezeptsuche weitergeleitet, nun kann der Benutzer ein eigenes Gericht auswählen. In beiden Fällen wird für das final gewählte Gericht eine Berechnung gestartet um die Nährwerte zu erhalten. Diese Nährwerte werden dem Benutzer zusammen mit gewählten Gericht im Frontend angezeigt, mit diesem Schritt endet der Programmablaufplan. Selbstverständlich hätte der „FoodTinder“-Algorithmus auch die Möglichkeit selbstständig dem Benutzer bereits im Auswahlscreen Nährwerte anzuzeigen. Da darauf in dieser Arbeit kein Einfluss genommen werden kann, wird in dieser Arbeit vom einfachsten Fall ausgegangen, dass der Algorithmus dies eben nicht tun.

4 Projektumsetzung

Die folgenden Kapitel dokumentieren die einzelnen Unterbereiche des gesamten Projekts.

4.1 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 Herrn Manuel Ullmann, sowie die Berechnung der Nährwerte der Rezepte in der kochbar-Datenbank. In der uns verfügbaren Version akzeptiert die Software als Aufrufparameter sogenannte. raw Dateien. Diese Dateien beinhalten das eigentliche Rezept das von der Software verarbeitet werden soll um schlussendlich die Nährwerte für das Gericht zu erhalten. Die Art, wie Herr Ullmanns Software Daten erhält und verarbeitet stellte sich als problematisch für das Projekt heraus. Hauptproblematik war dabei, dass die vorliegenden. raw Dateien prin-

ziell gecrawlte Webseiten beinhalten, jede .raw Datei repräsentiert dabei eine Rezeptseite von www.chefkoch.de. Die Software durchsucht diese Datei nach bestimmten HTML Tags und wertet diese aus um Inhaltsangaben, Mengenangabe, Rezepttitel und andere Daten über gewählte Gericht zu erhalten. Der uns gegebene Datensatz, also 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, so dass diese auch json Dateien als Eingabeparameter annimmt und verarbeiten kann. Um schlussendlich die Berechnung der Nährwerte der Einträge in der kochbar-Datenbank zu realisieren, folgt diese 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

4.1.1 Export der kochbar-Datenbank als json

Zum Export der kochbar-Datenbank wurde die Software MySQL Workbench⁸ verwendet. Das bereitgestellte Datenbankdump wurde dabei zur Bearbeitung lokal gehostet. Die entstandene Datenbank bietet sehr viel mehr Informationen als nötig. Eine vollständige Dokumentation der Datenbank, derer Tabellen und dessen Relation kann der Seminararbeit von Herrn Ullmann entnommen werden⁹. Zur Berechnung der Nährstoffe eines Gerichts benötigt die Software von Herr Ullmann lediglich dessen Zutaten. Die Zutaten, zusammen mit einem eindeutigen Identifier, eines Rezepts werden in der kochbar-Datenbank in der Tabelle **kochbar.recipes_analysis_recipe** dargestellt. Der ehemalige

⁸ MySQL Workbench, Download: <https://www.mysql.com/de/products/workbench/>, Abgerufen: 10.03.2018

⁹ Seminararbeit, Manuel Ullmann: <https://elearning.uni-regensburg.de/mod/resource/view.php?id=915461>

Link des Rezeptes¹⁰ wird dabei als eben genannter Identifier genutzt. Diese Identifier musste ebenfalls mit exportiert werden, um so in späteren Schritten eine Zuordnung der berechneten Nährwerte zu den ursprünglichen Rezepten zu ermöglichen. Schlussendlich beinhaltet der Export also die href¹¹ 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.

```
SELECT recipe_href,ingredients_string FROM newschema.kochbar_analysis_recipe limit 30000;
```

Abbildung 3: Query zum Export der Rezepte

Die Limitierung auf mehrere, kleine json Files hat auch den Vorteil, das im späteren Verlauf, bei der Verwendung 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 Repository¹², entnommen werden.

4.1.2 Anpassung 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 Nährwerte¹³ erfolgte über den Aufruf der **findnutr.py** Datei mit einer Rezeptdatei. Die **findnutr.py** importierte bei ihrem Aufruf die **readprep.py** Datei. Da diese importierte Datei für das Projekt keinen Nutzen hat, wurde der Import deaktiviert. Im Ursprungsprogramm war die

¹⁰ Wie bereits erwähnt handelt es sich bei den Rezepten um Webseiten der Seite www.chefkoch.de

¹¹ Eigentlich handelt es sich 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

¹² GitHub-Repository: <https://github.com/JakobFehle/KI4SG>

¹³ Die Software hat noch verschiedene andere Funktionen, in dieser Arbeit nutzten wir jedoch ausschließlich die Funktion zur Berechnung der Nährwerte.

readprep.py dafür zuständig aus der Rezeptdatei die Rezeptbeschreibung und die Anleitung zum Kochen zu parsen. Für die Berechnung der Nährwerte sind diese Informationen nicht von Belangen. Im Anschluss musste die Software json tauglich gemacht werden, dazu wurde die Rezeptklasse in **findnutr.py** angepasst werden. Die Klasse wird nun mit einem json Objekt instanziiert, aus welchem diese die entsprechenden Informationen zur Berechnung der Nährwerte erhält. Um dies zu ermöglichen wurde die Funktion „**__readJson**“ zum Einlesen einer json Datei, die Funktion „**__parseJson**“ zum Parsen der eingelesenen json Datei, sowie die Funktion „**__returnJson**“ zum Aufruf der Nährwertberechnung und dem Export des Ergebnisses als json Datei,¹⁴ implementiert. Der Export als json Datei ist dabei nicht zwingend, sondern nur zu aktivieren, falls dies vom Anwender gewünscht ist. Das Ergebnis wird abschließend in einem Array gespeichert. Die entstandene **findnutr.py** kann ebenfalls im GitHub-Repository gefunden werden.

4.1.3 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 vorliegende 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. Des Weiteren überprüft die **writeToDb.py** ob die passende Tabelle auf der Datenbank bereits existiert und legte diese, für den Fall das es diese noch nicht gibt, an. Die eigentliche Berechnung wird nun durch den Aufruf der **findnutr.py** gestartet. Dabei erwartet das Python Skript nun ein oder mehrere Rezepte im json Dateiformat. Die Nährwerte der entsprechenden Gerichte werden berechnet und via **writeToDb.py** in 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 Weise können auch neue Rezept zur Datenbank hinzugefügt werden, dabei müssen die Anwender lediglich wie oben beschrieben das **findnutr.py** Skript mit einer passenden json Datei aufrufen. Die Formatierung der json Datei kann dabei der im GitHub-Repository zu findenden **test.json** entnommen werden.

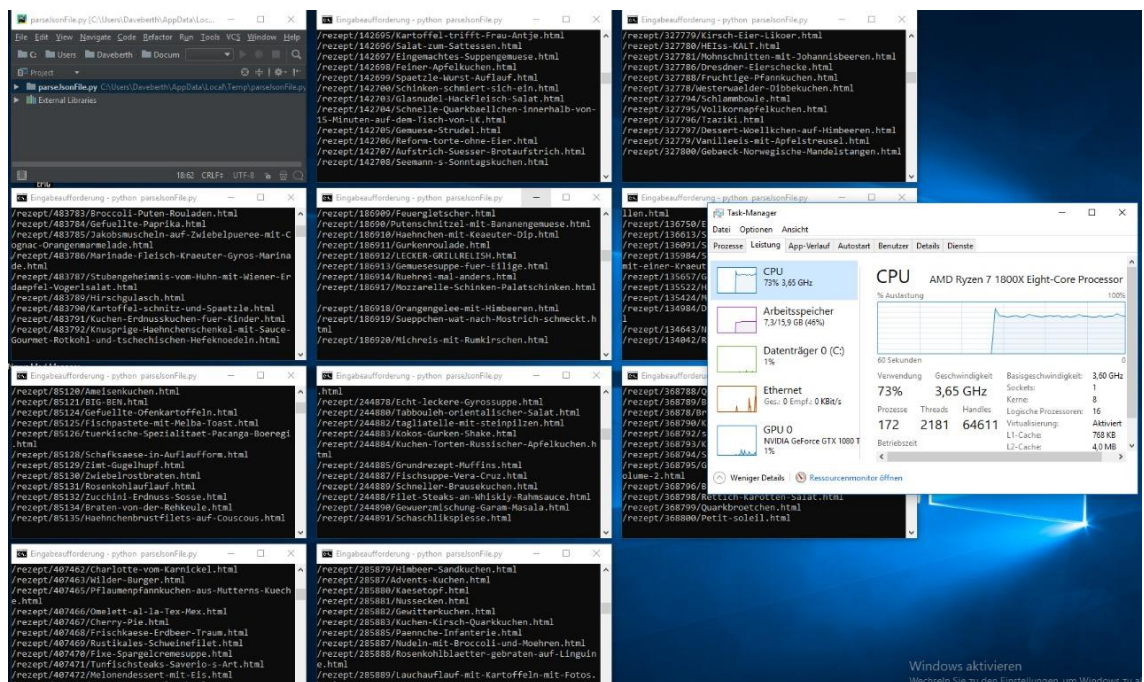


Abbildung 4: Reimport der Rezept-Nährwert-Kombination

Nach dem Reimport der Rezepte musste in einem letzten Schritt die eben angelegte Tabelle noch als SQL-Dump exportiert werden, um später auf dem Universitätsserver integriert werden zu können

4.2 Entwicklung Frontend zur Rezeptsuche

4.3 Entwicklung User Interface

5 Zusammenfassung und Ausblick

Dieses Kapitel soll die vorliegende Arbeit noch einmal zusammenfassen und zusätzlich einen Ausblick auf mögliche zukünftige Arbeiten geben. Es konnte gezeigt werden, wie mit Hilfe der von Herrn Ullmann entwickelten Software circa 300000 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 einfach direkten Zugriff auf die entsprechenden Werte, ohne diese selbstständig berechnen zu müssen. Additionally konnten in dieser Arbeit demonstriert werden wie Anwender zukünftig auch Rezepte bewerten können ohne, dass diese zwingend dem Repertoire der Internetseite 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. Selbstverständlich, wie ist Softwareentwicklung wohl meistens ist, haben sich während der Projektdurchführung einige Schwachstellen und konzeptionelle Fehler mit 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.

5.1 Optimierung der Datenbanksuche

In der aktuellen Version, des in dieser Arbeit entwickelten, User Interface gibt es nicht die Möglichkeit die Datenbank nach Rezeptbestandteilen beziehungsweise nach bestimmten Zutaten zu durchsuchen. Während des Suchvorgangs wird lediglich die Spalte des Rezepttitels, also der eindeutige Identifier, des Gerichts beziehungsweise 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 können oder diese, falls gewünscht, auch ausschließen können, ohne, dass diese zwingend im Namen des Gerichts vorkommen. Die Entwicklung einer performanten Volltextsuche der Inhaltsstoffspalte für die kochbar-Datenbank wäre eine mögliche Thematik für zukünftige Arbeiten.

5.2 Optimierung des Datenbankdesigns

Die vorliegenden, von uns entwickelte Tabelle, also die Tabelle die die Zuordnung von Rezepttitel und Nährwerte enthält, verzichtet auf die Verwendung von traditionellen Datenbankattribute. So wird in der Tabelle keine Referenz auf die eigentlich ursprüngliche Rezepttabelle angelegt. Klassischer weiße sollte hier der Titel der Rezepte in der Tabelle als FOREIGN KEY gesetzt werden, 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.

5.3 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 Parse von Mengenangaben zu haben die durch Abkürzungen definiert sind wie etwa „60 gr.: Mehl“. Schlussendlich konnte, durch den Einsatz eines Eventhandlers in der `parseJsonFile.py`, die Problematik umgangen werden. Rezepte die nicht eindeutig bewertet werden konnten wurden nicht in die resultierende Tabelle aufgenommen, sondern ausgelassen. Die Summe der so verlorenen Rezepte beläuft sich bei etwa 1% des gesamten kochbar-Datensatz, so konnten circa 5000 Rezepte nicht bewertet werden. Für zukünftige Arbeiten wäre die Verbesserungen der „Parse -Quote“ selbstverständlich interessant, vor allem auch um individueller auf Benutzereingaben von neuen Rezepten reagieren zu können.

6 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