



29.4.2019

Projektdefinition

MealFinder

Erstellt von: Marc Budde, Marvin Juwig

Inhalt

1	Hintergrund	2
2	Problemstellung	2
3	Ziele	2
4	Umfang	2
5	Vorgehen	3
6	Risiken	3
7	Randbedingungen	3
8	Kritische Erfolgsfaktoren	3
9	Benötigte Ausrüstung und Einrichtungen	4

1 Hintergrund

Das Projekt wird im Rahmen der Veranstaltung Webbasierte Anwendungen 2 unter Leitung von Prof. Dr. Kristian Fischer an der TH Köln, Campus Gummersbach realisiert.

2 Problemstellung

Es sind noch Lebensmittel im Kühlschrank übrig und es ist nicht bekannt, was aus diesen an Gerichten zubereitet werden könnte. Welche Nährwerte weisen die Lebensmittel bzw. Gerichte auf?

3 Ziele

Eine Software-Anwendung zu realisieren mit der man Mahlzeiten auf Basis der Lebensmittel, die man gerade zu Verfügung hat, auswählen kann. Dazu gibt es folgende Unterziele:

Verbindlich:

- Eingabe der vorhandenen Lebensmittel (durch Dienstnutzer)
- Die Ausgabe der Gerichte soll auf die Anzahl der fehlenden Zutaten beschränkt werden können
- Gerichtsvorschläge werden auf Basis der angegebenen vorhandenen Lebensmittel ausgegeben
- Nährwerte von Lebensmitteln oder Gerichten können ausgegeben werden.

Optional:

- Filtereinstellungen (z.B. vegan, vegetarisch)

4 Umfang

Für das Projekt werden erstmal alle vorgesehenen verbindlichen Ziele realisiert. Optional könnte das System noch erweitert werden.

5 Vorgehen

Realisiert wird das Projekt wie folgt:

Zuerst muss ein Domänenmodell erstellt werden. Anschließend wird die Rest-Modellierung fertig gestellt. Außerdem ist ein Dienstanbieter zu implementieren und mindestens ein externer Webservice eingebunden. Um eine Kommunikation mit dem Dienstanbieter zu ermöglichen wird ein Dienstanutzer integriert und dessen Anwendungslogik implementiert.

6 Risiken

Risiko	Mögliche Auswirkung auf den Projekterfolg
Kein Zugriff auf Gericht/Nährwerte API	Anwendung kann nicht umgesetzt werden im Rahmen der Vorgaben
Nährwert/Gerichte werden nicht korrekt ausgegeben	System wird unbrauchbar bzw muss korrigiert werden

7 Randbedingungen

Das System wird mithilfe von Node.js umgesetzt.

8 Kritische Erfolgsfaktoren

- Das System muss die Mindestanforderungen erfüllen. Darunter fällt, dass mindestens ein externer Webservice (Dienstgeber, Api...) eingebunden wird. Außerdem soll die Server-Client Kommunikation einwandfrei funktionieren, damit interne Berechnungen, sowie Ein- und Ausgaben korrekt ausgegeben bzw bearbeitet werden.
- Das System muss ausführbar und damit zu testen sein.
- Der Code muss sinnvoll strukturiert, sowie kommentiert und in GitHub bereitgestellt werden.

9 Benötigte Ausrüstung und Einrichtungen

- Entwicklungsumgebung
- Externer Webservice
- PC/Laptop