

Ausarbeitung

Entwurf einer Datenbank für ein Verwaltungssystem von Lebensmitteln in einem Kühlschrank

Private Hochschule für Wirtschaft und Technik Vechta/Diepholz

vorgelegt von: Claas Meints

Semester: 6

Inhaltsverzeichnis

1 Beschreibung der Anforderungen	1
2 Modellierung der Datenbank	4
3 Entwicklung des Backends	5
4 Entwicklung und Simulation des Frontends	6
5 Ergebnisse	7
Glossary	8
Acronyms	9
Glossar	10

© 2022

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

1 Beschreibung der Anforderungen

1.1 Das Kühlschrank-Verwaltungssystem

Die im Rahmen dieser Ausarbeitung zu entwerfende Datenbank ist Teil eines Projektes zur Entwicklung eines Verwaltungssystems für Lebensmittel, die in Kühl- oder Gefrierschränken gelagert werden. Ziel des Projektes ist es, eine günstige Alternative zu smarten Kühlschränken zu entwickeln, die es ermöglicht die Lagerung von Lebensmitteln zu überwachen, ohne einen neuen Kühlschrank anschaffen zu müssen. Mit Hilfe der Lösung soll der Verschwendung von Lebensmitteln entgegengewirkt werden.

In vielen Haushalten werden aufgrund mangelnder Planung bzw. mangelndem Überblick über die Situation im Kühlschrank teilweise große Mengen an Lebensmitteln entsorgt. Zum Einen geraten Lebensmittel die im Kühlschrank weiter hinten stehen in Vergessenheit und werden nicht rechtzeitig verwendet. Zum Anderen werden Lebensmittel gekauft, ohne dass klar ist, wann und in welcher Form dieses Benötigt werden. Die zu entwickelnde Lösung soll beiden Problemen entgegenwirken. Durch die Überwachung der Lebensmittellagerung wird die Planung von Einkauf und Verwertung dieser erleichtert.

Aus Benutzer:innen Sicht stellt eine Smartphone-App das zentrale Element der Lösung dar. Über diese können Kühl- und Gefrierschränke, oder andere Lagerstätten für Lebensmittel eingebunden und verwaltet werden. Auf einen Blick sollen Benutzer:innen alle Lebensmittel in all ihren Geräten aufgelistet bekommen. Zusätzlich soll der Füllstand bzw. die Menge der jeweiligen Lebensmittel und die Haltbarkeit angezeigt werden. Benutzer:innen sollen zusätzlich die Möglichkeit haben detaillierte Informationen wie Zutatenlisten einzelner Produkte ab zu rufen. Mit Hilfe der App können Benutzer:innen darüber hinaus ihren Einkauf planen und nicht automatisierbare Ein- bzw. Auslagerungsprozesse manuell einpflegen.

Damit Benutzer:innen die Lösung tatsächlich verwenden muss die Bedienung so komfortabel wie möglich sein. Besonders kritisch ist die Anzahl der manuell zu pflegenden Prozesse. Lebensmittel sollen mit Hilfe eines IoT-Gerätes automatisch Ein- bzw. Ausgelagert werden können. Der Großteil an Produkten wird mit Hilfe eines Barcode-Scanners

erfasst. Produkte ohne Barcode sollen mittels bildgestützter Verfahren erkannt werden. Um die IoT-Geräte optimal einbinden und verwalten zu können sollte die Lösung eine IoT-Plattform integrieren.

Bei der Entwicklung der Lösung werden spätere potentielle Erweiterungen mit bedacht. Unter dem Aspekt der Lebensmittelverschwendung wäre die Integration einer Rezeptverwaltung bzw. einer Ernährungsplanung besonders nützlich. Auch die Analyse von Nutzungsdaten kann potentiell dazu Verwendet werden die Zahl der nötigen manuellen Eingriffe von Benutzer:innen zu reduzieren. Abseits der Anwendungsfälle des Lebensmittelverwaltungssystems, kann eine doppelte Nutzung der in Kühlschränken installierten IoT-Geräte sinnvoll sein. So könnten zum Beispiel Temperaturdaten von Kühl- und Gefrierschränken interessant für Projekte sein, die den Energieverbrauch von Haushalten optimieren.

Die im Rahmen dieser Ausarbeitung zu entwickelnde Lösung stellt das MVP (Minimum Viable Product), also das minimal funktionsfähige Produkt des gesamten Verwaltungssystems dar. Der Fokus liegt dabei auf dem Entwurf der Datenbank, sowie den zum Test der Datenbank benötigten Komponenten. Es wird Prototyp für das Backend in Node.js entwickelt, dass an die Datenbank angebunden ist. Dieses enthält die nötige Verwaltungslogik des Systems und bietet eine REST-API für die Anbindung der Smartphone-App. Über die App wird es in der MVP-Phase nur möglich sein die Liste der Aktuell eingelagerten Lebensmittel an zu zeigen.

1.2 Anforderungen an die Datenbank

Aus dem beschriebenen Projektziel ergeben sich die Anforderungen an die zu entwerfende Datenbank. Da die Daten unterschiedlicher Benutzer:innen durch ein Zentrales Backend verarbeitet werden wird eine Benutzer:innenverwaltung benötigt. Die Datenbank muss also die Möglichkeit bieten Benutzer:innen inklusive ihrer Anmeldedaten zu speichern. Darüber hinaus müssen allen Benutzer:innen ihre Geräte (also Kühl- oder Gefrierschränke, etc.) sowie ihre Einkaufslisten zugeordnet werden. Alle nicht einzelnen Benutzer:innen zuordbaren Entitätsklassen sollen möglichst übergreifend genutzt werden.

- Speicherung der Produkte
- Benutzerverwaltung
- Geräteverwaltung
- (Automatisches) Erstellen von Einkaufslisten

- Einordnen der Produkte in Kategorien
- Speicherung von Produkthistorie -> nützlich für Einkaufslisten und um zu erkennen ob Produkte neu gekauft sind oder wieder zurückgelegt wurden
- Informationen über Haltbarkeit und Füllstand der Produkte
- Bilder und Icons für das Frontend
- Erweiterbarkeit für weitere Funktionen mitdenken:

1.3 Abgrenzung der Anforderungen

- MVP!
- Keine voll Funktionsfähige/Ansprechende App
- Keine Verbindung zur IoT-Plattform
- Keine Anbindung echter Geräte -> Simulation durch Python Skript
- Kein Deployment in der Cloud
- Keine Benutzermanagement -> keine Authentifizierung / Autorisierung

2 Modellierung der Datenbank

2.1 Konzeptionelles Schema

2.2 Physisches Schema

2.3 Nachweis der Normalformen

3 Entwicklung des Backends

3.1 Entwurf der Architektur

3.2 Schnittstellendefinition

3.3 Implementierung des Datenbankmodells

4 Entwicklung und Simulation des Frontends

4.1 Entwurf der Android-App

4.2 Simulation von Drittsystemen

5 Ergebnisse

Glossar

Entitätsklasse . 2

Acronyms

API Application Programming Interface. 10

MVP Minimum Viable Product. 2

REST Representational State Transfer. 10

Glossar

REST-API bla bla bla. 2

Backend bla bla bla. 2