Bei dieser Präsentation werden zunächst die ursprüngliche Vision rekapituliert, danach die letzten Artefakte präsentiert und im Anschluss der vertikale Prototyp. Zum Schluss gibt es einen Ausblick, was für die Weiterentwicklung des Projekts denkbar wäre.

Vision am Anfang

Ein interaktives System zur Umverteilung von Lebensmitteln für private Haushalte

Alleinstellungsmerkmale:

- Private Gruppen ermöglichen
- Automatische Bewertung von Inseraten
- Informationen zur richtigen Lagerung

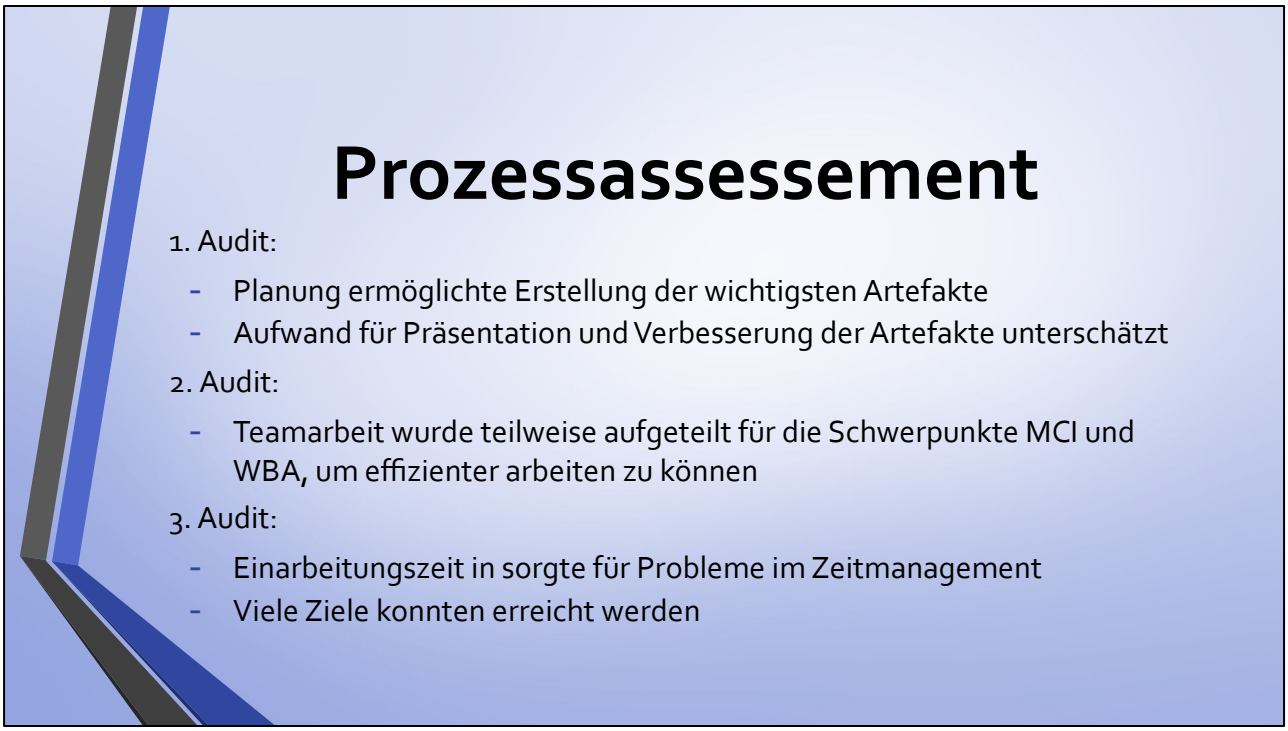
Das Ziel des Projektes war ein interaktives System zur Umverteilung von Lebensmitteln zu entwickeln. Private Haushalte sollten die Möglichkeit bekommen sich miteinander zu vernetzen und nicht mehr benötigte Produkte anderen zur Verfügung zu stellen. Womit die allgemeine Verschwendung von Lebensmitteln reduziert werden sollte.

Im Vergleich zu den anderen Anbietern auf dem Markt, soll unser System folgende Möglichkeiten bieten:

- Private Gruppen ermöglichen:
  Es gibt verschiedene Gründe warum ein Nutzer sein Produkt nicht mit jedem teilen möchte, deshalb soll es ihm ermöglicht werden, den Angebots- und Suchradius auf bestimmte Gruppen zu beschränken. So genießen sie das gewohnte von sozialen Netzwerken, aber sind dennoch in der Lage auch öffentliche Inserate zu betrachten.

- Automatische Bewertung von Inseraten:
  Andere Systeme bieten keine Möglichkeit Angebote bewerten zu lassen nach ihrer Qualität und Entfernung zum Suchenden. In diesem Projekt sollen Anbieter die Möglichkeit bekommen, ihr Produkt durch einfache Angaben automatisiert bewerten zu lassen, damit diese bei der Suche hervorgehoben werden können. Hierdurch wird ebenfalls ermöglicht, das Produkte die bereits zu lange angeboten werden automatisch gekennzeichnet oder gelöscht werden.

- Informationen zur richtigen Lagerung:
  Um Suchenden dabei zu helfen ihr erhaltenes Produkt möglichst einwandfrei aufzubewahren, sollen diese auf Wunsch, Informationen zur richtigen Lagerung erhalten, wenn sie das Inserat betrachten.



Als die angedachte Idee für das Exposé abgesegnet wurde, war die erste Aufgabe die gemeinsame Planung eines Projektplans. Dieser umfasste den groben Zeitplan und die Artefakte die bis zum 1. Meilenstein als wichtig erachtet wurden.

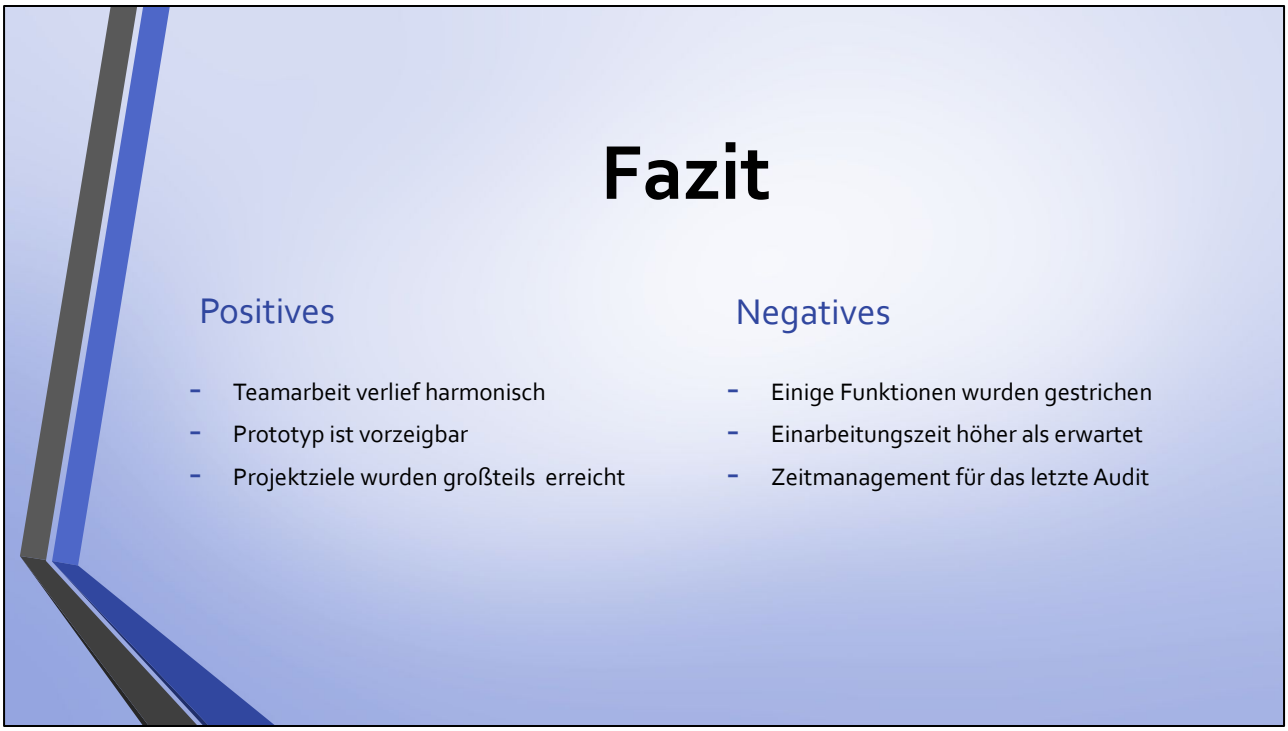
Durch die großzügige Planung konnten die wichtigsten Artefakte ohne Probleme erstellt werden. Lediglich die Zeit für die Vorbereitung der Präsentation wurde etwas knapp bemessen.

Nach der Kritik aus dem 1. Audit wurden bisherige Artefakte und das weitere Vorgehen evaluiert und angepasst und die nötigen Schritte für die nächsten angesetzten Meilensteine geplant. Bei der Planung wurden kleinere Meilensteine angestrebt, die jeweils zum Ende der Woche fertig zu stellen waren.

Hier wurde die Schwerpunkte der MCI und WBA unter dem Team aufgeteilt, wobei das jeweils andere Teammitglied trotzdem Einfluss und Zugriff auf alle Informationen hatte und Feedback und Verbesserung geben konnte. Als die Entwicklung der Rapid Prototypen anstand, hat sich ein Mitglied darauf konzentriert und das andere die restlichen Artefakte übernommen.

Für die Vorbereitung der Präsentation des 2. Audits wurde ausreichend Zeit eingeplant, und es kam hierdurch zu einer früheren Fertigstellung, wodurch die Abgabe nochmal in Ruhe kontrolliert und verbessert werden konnte.

Die letzte Iteration kostete viel Einarbeitungszeit in die Entwicklungsumgebung und Verbindungsprobleme zwischen Client/Server erschwerten das Vorankommen. Es musste am Ende abgewägt werden, welche Funktion für den finalen vertikalen Prototypen als wichtig erachtet wurden. Um einen lauffähigen Prototyp präsentieren zu können, wurde sich auf Kernfunktionen des Anbieten/Suchen und die Alleinstellungsmerkmale konzentriert.



Zu Beginn des Projektes wurden Ziele und Anforderungen für das Projekt definiert, die für das angedachte System von Bedeutung sind. Diese Ziele wurden zu einem großen Teil erfüllt, auch wenn in Teilbereichen der Implementierung einige Features gekürzt wurden.

Der Prototyp ist in der Lage Inserate zu erstellen und sich alle Inserate, die auf dem Server gespeichert sind, anzeigen zu lassen. Ebenfalls können Inserate bereits als privat markiert werden, wodurch nur Nutzer einer bestimmten Gruppe dieses Inserat angezeigt bekommen. Bei der Erstellung berechnet der Server eine interne Bewertung anhand der eingegebenen Daten und speichert diese mit dem Inserat.

Außerdem besteht eine Komfortfunktion, die den Standort und das aktuelle Wetter in der Umgebung des suchenden Nutzers überprüft und ihn über schlechte Wetterbedingungen informiert. Der Nutzer hat dann die Möglichkeit den Suchradius für Inserate zu reduzieren.

Durch Schwierigkeiten bei der Einarbeitung in die Entwicklungsumgebung mussten einige Features gekürzt werden. So funktioniert der Prototyp bisher lediglich mit einem bestimmten Nutzer. Die Funktion sich zu registrieren und authentifizieren könnte im Nachhinein implementiert werden. Durch das Fehlen der Nutzer wurden auch auf Features verzichtet die zur Verwaltung mit den Inseraten im Zusammenhang stehen.

Sollten die fehlenden Features nachträglich eingebaut werden, würde das Projekt eine gewisse Marktreife entwickeln. Weiterhin könnten das Verhalten zur Nutzung analysiert bzw. evaluiert werden, die daraus resultierenden Wünsche und Feedback zur weiteren Planung genutzt werden.

Für das Plakat wurden sich auf die wesentlichen Funktionen des funktionierenden Prototypen konzentriert. Die Funktionen wurden kurz beschrieben und die Screenshots von der entwickelten Anwendung geben einen direkte Überblick auf das System.

Die Farben der TH Köln wurden als Hintergrund für die Überschriften und die Zierleiste genutzt. Als weitere Akzentfarbe wurde die von der Anwendung genutzte Hauptfarbe verwendet.

Im folgenden wird der vertikale Prototyp vorgeführt. Der zugehörige Code befindet sich im Github.

Zum Start wird dem Server eine GET Anfrage geschickt für alle Inserate die aus privaten Gruppen des Nutzers stammen. Ebenfalls eine weitere GET Anfrage für die die vorhandenen Kategorien. Bei erfolgreicher Antwort werden beide auf der Startseite dargestellt. Während die Daten geladen werden, wird eine Ladeanimation angezeigt, um den Nutzer über den Umstand zu informieren.

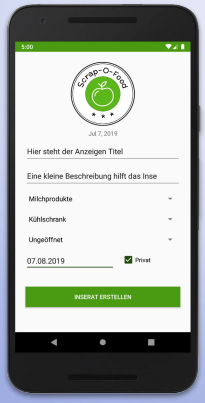
Oberhalb der Anwendung befinden sich die privaten Inserate und unterhalb die Kategorien - mitsamt der Bilder.

Die privaten Inserate leiten den Nutzer direkt zur zugehörigen Detailseite, die Kategorien leiten zu der passenden Kategorie die alle öffentlich erstellten Inserate darstellt, für die jeweilige Kategorie.

Auf der Kategorieseite wird oberhalb eine Box dargestellt die den Nutzer über richtige Lagerung der Lebensmittel informiert. Diese wird bei Berührung durch einen Popup erweitert. Ebenfalls wird beim Abrufen der Kategorie zunächst eine Standort und Wetterabfrage gesendet, wenn das Wetter über gewisse Grenzbereiche stößt, wird der Nutzer über diesen Umstand mittels Alertdialog informiert und bekommt die Wahl, die Suche auf näher liegende Inserate zu beschränken. Der GET Anfrage zur Darstellung der Inserate werden dann die Standortdaten des Nutzers mitgegeben, wenn diese ermittelt wurden, und der Server filtert die Anzeigen.

Die Detailseite beinhaltet alle wichtigen Informationen, die der Nutzer bei der Bestellung eingegeben hat. Da der Bilderupload in diesem Prototyp noch nicht implementiert wurde, werden lizenzfreie Stockbilder angezeigt. Über die Detailseite kann der Anbieter direkt über Mail oder Telefon kontaktiert werden.

Inserat erstellen




```
app.post('/meals', function (req, res) {
    let mealsIndex = meals.length;
    let newMeal = {
        "idMeal": mealsIndex,
        "idUser": req.body.idUser,
        "idGroup": req.body.idGroup,
        "strMeal": req.body.strMeal,
        "strDescription": req.body.strDescription,
        "strLongitude": req.body.strLongitude,
        "strLatitude": req.body.strLatitude,
        "adDate": req.body.adDate,
        "strCategory": req.body.strCategory,
        "strStoring": req.body.strStoring,
        "strCondition": req.body.strCondition,
        "strMHD": req.body.strMHD,
        "strMealThumb": "http://" + serverIP + ":" + globalPort + "/upload" + req.body.strCategory + picFormat,
        "strMail": req.body.strMail,
        "strPhone": req.body.strPhone,
        "rating": createRating(req.body.strCategory, req.body.strStoring, req.body.strCondition)
    };

    newMeal.strMHD = getMHD(newMeal.strMHD, newMeal.rating);
    meals.push(newMeal);
    res.status(201).json(newMeal);
});
```

Server-Side

Der Nutzer erstellt die Anzeige und gibt alle nötigen Informationen zu seinem Produkt ein. Er bekommt die Möglichkeit das Inserat direkt privat zu markieren, sodass es nur für Nutzer innerhalb dieser Gruppe zur Verfügung steht. Ist der Nutzer zufrieden sendet er eine POST Anfrage an den Server.

Der Server erhält die Anfrage und ergänzt fehlende Informationen, zum Beispiel die Inserat ID und ein für die Kategorie passendes Bild. Die Möglichkeit eigene Bilder hochzuladen würde in einer späteren Iteration hinzugefügt.
Ebenfalls analysiert der Server die eingegebenen Daten bezüglich Kategorie, Zustand und Lagerung. Er berechnet daraus eine Wertung für das Inserat und ein Ablaufdatum der Anzeige.

REST im System

```
@GET("categories")
Call<Categories> getCategories();

@GET("meals")
Call<Meals> getMealByCategory(@Query("c") String category,
                              @Query("lat") double lat,
                              @Query( "long") double lon);

@GET("meals")
Call<Meals> getMealByCategory(@Query("strCategory") String category,
                              @Query("idGroup") int idGroup);

@GET("meals")
Call<Meals> getMeal(@Query("idGroup") int idGroup);

@GET("meals")
Call<Meals> getMealByName(@Query("strMeal") String strMeal);

@POST("meals")
Call<Post> saveMeal(@Body Post post);
```

Client-Side

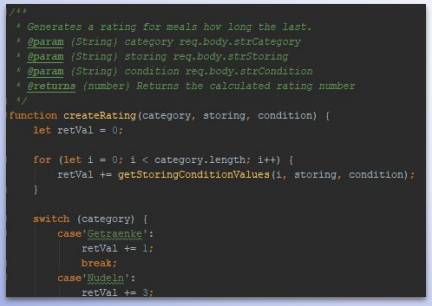
```
if (req.query.strCategory != null) {
    for (let i = 0; i < meals.length; i++) {
        tempMeals.meals = meals.filter(function (meals) {
            return meals.strCategory === req.query.strCategory
        });
    }
}
if (req.query.idGroup != null) {
    for (let i = 0; i < meals.length; i++) {
        if (tempMeals.meals.length > 0) {
            tempMeals.meals = tempMeals.meals.filter(function (meals) {
                return meals.idGroup === req.query.idGroup
            });
        } else {
            tempMeals.meals = meals.filter(function (meals) {
                return meals.idGroup === req.query.idGroup
            });
        }
    }
}
```

Server-Side

Hier sind Beispiele zu finden, die der vertikale Prototyp an REST Befehlen im Server ausführen kann.

So werden “Categories” über eine GET Anfrage abgefragt, “Meals” werden über verschiedene Queries abgefragt. Ebenfalls werden neue Anzeigen in “Meals” über eine POST Anfrage erstellt und angelegt.

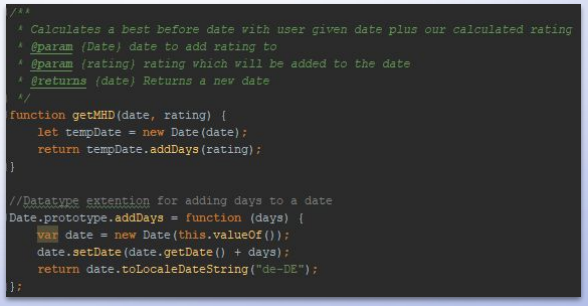
Anwendungslogik



```
/**
 * Generates a rating for meals how long the last.
 * @param {String} category req.body.strCategory
 * @param {String} storing req.body.strStoring
 * @param {String} condition req.body.strCondition
 * @returns {number} Returns the calculated rating number
 */
function createRating(category, storing, condition) {
    let retVal = 0;

    for (let i = 0; i < category.length; i++) {
        retVal += getStoringConditionValues(i, storing, condition);
    }

    switch (category) {
        case'Getraenke':
            retVal += 1;
            break;
        case'Nudeln':
            retVal += 3;
```



```
/**
 * Calculates a best before date with user given date plus our calculated rating
 * @param {Date} date to add rating to
 * @param {rating} rating which will be added to the date
 * @returns {date} Returns a new date
 */
function getMHD(date, rating) {
    let tempDate = new Date(date);
    return tempDate.addDays(rating);
}

//Datatype extention for adding days to a date
Date.prototype.addDays = function (days) {
    var date = new Date(this.valueOf());
    date.setDate(date.getDate() + days);
    return date.toLocaleDateString("de-DE");
};
```

Links:
Berechnung des internen Ratings. Anhand dieses Ratings werden die Anzeigen sortiert. Je höher der Wert, desto länger ist dieses Produkt haltbar.

Rechts:
Das vom Benutzer angegebene MHD wird verwendet um ein neues Datum mithilfe des von uns kalkulierten Ratings zu generieren.

Ausblick

- Optimierung von Performance
- Implementierung der fehlenden Funktionen
- Analyse und Evaluation von Nutzungsverhalten
- Mehr Nutzer für das System gewinnen
- Sponsoren und Partner für den Vertrieb finden

Im Anschluss an das nun abgeschlossene Projekt könnte dieses Weiterentwickelt werden und so eine gewisse Marktreife entwickeln.

Es sollten zunächst Performance verbessert und Ladezeiten reduziert werden, um den Nutzer die Anwendung zugänglicher zu machen. Ebenfalls müssen für eine Veröffentlichung die fehlenden Funktionen implementiert werden, zum Beispiel die Registrierung und Authentifizierung und die Verwaltung von Inseraten.

Im Anschluss können weitere Analysen und Evaluation durchgeführt werden, um das System weiter zu optimieren und an die Wünsche der Nutzer anzupassen.

Sobald die Anwendung veröffentlicht wurde, gilt es die Reichweite zu erhöhen und mehr Nutzer für das System zu gewinnen. Denkbar wäre auch die Zusammenarbeit mit Sponsoren und Partner für den Vertrieb für Server und Marketing.

Vielen Dank für die Aufmerksamkeit.