

Fazit

Dieses Dokument soll die Eigenschaften der bis zum 18.01.2016 erreichten Implementation den formulierten Projektzielen entgegenstellen und den Zielerreichungsgrad kritisch diskutieren.

Inhaltsverzeichnis

FAZIT	1
ZIELERREICHUNGSGRADE DER PROJEKTZIELE	1
ANALYSE DER ANFORDERUNGSERFÜLLUNG	2
FUNKTIONALE ANFORDERUNGEN	2
QUALITÄTSANFORDERUNGEN	3
ORGANISATIONALE ANFORDERUNGEN	3
ZWISCHENFAZIT ANFORDERUNGSERFÜLLUNG	3
ZIELERREICHUNGSGRADE BEI DER UMSETZUNG DER ALLEINSTELLUNGSMERKMALE	3
VERTIKALER PROTOTYP	4
IMPLEMENTIERTER KOMMUNIKATIONSABLAUF	4
AUS ZEITLICHEN GRÜNDEN AUSGELASSEN	5
AUSBLICK	5

Zielerreichungsgrade der Projektziele

Die strategischen Ziele des Projektes lassen sich noch nicht unmittelbar auf Erfüllung prüfen, die Wahrscheinlichkeit ihrer Erfüllung kann allerdings bereits jetzt abgeschätzt werden. Die dort formulierten muss-Ziele „Lebensmittelverschwendung reduzieren“ und „ungleiche Verteilung von Lebensmitteln bekämpfen“ können selbst mit dem Implementierten funktionsumfang zumindest prototypisch erreicht werden. Mithilfe des implementierten Benachrichtigungsmechanismus verfügt das System über Möglichkeiten zum Anstoßen anderer Teilnehmer, womit sich auch das soll-Ziel „mobilisieren für soziales Engagement im Hinblick auf Lebensmittelverschwendung und Verteilung“ zumindest in Angriff nehmen lässt.

Mit den an der REST-Schnittstelle modellierten, aber nicht mehr implementierten Statistiken einer Distributionsgruppe sollten die beiden letzten strategischen Ziele („Sensibilisieren für das Thema Umweltschutz“ und „Bildung und Zusammenhalt lokaler Gemeinschaften fördern“ erreicht werden. Da die Erstellung solcher Statistiken jedoch schon mit der Implementierung von Logging-Logik an diversen Stellen des Dienstgebers hätte erreicht werden können und dies nicht als Kern des Systems eingestuft wurde kann dieses Ziel zunächst nur umrissen werden.

Auf Ebene der taktischen Ziele konnte das Ziel „Anbieten einer Distributionsmöglichkeit zum anbieten und tauschen von Lebensmitteln auf lokaler bis regionaler Ebene“ auf Seiten des Dienstgebers, der Ressourcen zur Erstellung von Angeboten beinhaltet, implementiert werden. Die Entwürfe der Benutzungsschnittstelle zeigen außerdem bereits Screens zur Angebotserstellung in einer zweiten Iteration. Der zur

Demonstration der verteilten Anwendungslogik und Demonstration während der Projektpräsentation ausgewählte Use Case adressiert das zweite taktische Ziel (Organisieren von privaten Lebensmittelspenden und deren Transport zu den Tafeln). Dieses Ziel wird erfüllt, mit dem Implementierten Funktionsumfang können Benutzer in der Rolle des Sammlers ihre Situation analysieren lassen und darauf zugeschnittene Vorschläge zur Sammlung von Lebensmitteln erhalten. Mechanismen zur Organisation des Transportes gesammelter Lebensmittel zu den Tafeln wurde aus zeitlichen Gründen nicht mehr implementiert. Zur Messung des Zielerreichungsgrades von Zielen welche die Qualität des Systems als Werkzeug zur Aufgabenerledigung beinhalten muss in erster Linie die Erfüllung spezifizierter Anforderungen analysiert werden, was im folgenden Abschnitt geschehen soll.

Analyse der Anforderungserfüllung

Der bisherige Stand der Implementierung kann eine Reihe der spezifizierten Anforderungen nicht erfüllen. Dies liegt darin begründet, dass für eine umfangreiche Implementation der Benutzungsschnittstelle am Ende des Projektes keine zeitlichen Ressourcen mehr vorhanden waren. Zudem erscheint eine Implementierung der UI zu diesem Zeitpunkt nur für Demonstrationszwecke sinnvoll, da sich der UI-Prototyp erst in der zweiten Iterationsstufe befindet und eine zuvor eingeplante zweite formative Evaluation nicht mehr durchgeführt werden konnte.

Funktionale Anforderungen

Welche funktionalen Anforderungen durch Elemente der Benutzungsschnittstelle adressiert werden ist im Artefakt „UI-Prototyp“ erläutert und wird hier nicht mehr diskutiert. Dort nicht adressiert wurden die Anforderungen #3,#11#12,#14,#19,#23,#25,#26 , da diese sich weniger auf die UI-Gestaltungslösung als auch dem Benutzer verborgene Systemcharakteristika beziehen. Wie diese Anforderungen erfüllt werden soll hier kurz erläutert werden.

- #3 – Angebotszustände wurden als Attribut der Angebotsressource implementiert
- #11 & #12 – Mit der Implementierung des Nachrichtenaustauschs über RabbitMQ verfügt das System über die Möglichkeit Anbieter über diverse Sachverhalte zu Benachrichtigen
- #14 - Der Status einer Sammelaktion wurde ebenfalls als Attribut der Sammelaktionsressource implementiert
- Übernimmt ein Benutzer die Aufgabe eine Sammelaktion hinzuzufügen wird diese Mittels POST-Operation beim Dienstgeber angelegt, woraufhin Wird eine Sammelaktion werden auch im Prototypen
- #19 – Die Situationsanalyse übernimmt exakt diese Aufgabe, die Anforderung konnte damit erfüllt werden
- #23 – Mit dem Setup eigener OSRM-Instanzen und funktionierender Kommunikation zwischen diesen und dem Dienstgeber kann auch diese Anforderung erfüllt werden
- #25 – Der Algorithmus zur Erstellung von Teiltransporten, die jeweils genau eine Spende zu einem Ziel transportieren wird zum Zeitpunkt des Verfassens dieses Dokumentes noch implementiert, es ist nicht sicher ob dies noch implementiert werden kann.

- #26 – Mit der Anbindung eines (nur prototypisch implementierten) Dienstnutzers an die verwendete nachrichtenorientierte Middleware können Zuständige Tafeln nicht nur ermittelt, sondern auch über das vorhandene Spendenvolumen, Standorte und Sammelaktionen informiert werden.

Qualitätsanforderungen

Die Erfüllungsgrade der meisten qualitativen Anforderungen können in diesem Projektstadium noch nicht gemessen, dazu zählen bspw. das Verhalten des Systems bei 10.000 gleichzeitigen Benutzern (Anf. #57) und die Erreichbarkeit des Systems (Anf. #54). Die Anforderungen #58 - #61 beziehen sich auf in der Veranstaltung nicht fokussierte Eigenschaften des Systems (Sicherheit, Wiederherstellbarkeit von Datenbeständen usw.) und konnten bisher nicht erfüllt werden. Anforderungen #49-#53 werden im UI-Prototypen adressiert, können aber Aufgrund noch nicht durchgeführter Evaluationen zu diesem Zeitpunkt nicht sicher bewertet werden.

Organisationale Anforderungen

Sämtliche organisationalen Anforderungen werden bereits im Prototypen erfüllt. Mit der Auswahl der Android Plattform wird eine im Endverbrauchersegment weit verbreitete Plattform als Zielplattform gewählt, Anforderungen #62-64 werden damit erfüllt. Der Algorithmus zur Auffindung sinnvoller Sammelaktionen ist zum aktuellen Projektstand so implementiert, dass er Sammelaktionen immer innerhalb eines PLZ-Bereichs sucht, damit wird der Gebietsschutz in jedem Fall eingehalten und Anforderung #65 erfüllt. In Zukunft kann dieses Vorgehen mit relativ wenig Aufwand erweitert werden um PLZ-übergreifend agieren zu können. Mit der prototypischen Implementation des Tafel-Dienstnutzers als Web App ist dieser Teil des Systems Plattformunabhängig konzipiert und erfüllt damit Anforderung #67. Zur Erfüllung von Anforderung #66 wurden diverse Vorkehrungen an der Benutzungsschnittstelle getroffen. Damit sind alle organisationalen Anforderungen erfüllt.

Zwischenfazit Anforderungserfüllung

Es lässt sich feststellen, dass viele der für den Prototypen als relevant markierten Anforderungen erfüllt werden konnten. Da dies nur einen Ausschnitt aller ermittelten Anforderungen darstellt wird ebenfalls deutlich, dass sich das System zum Zeitpunkt der Abgabe erst in einer frühen Iterationsstufe befindet. Die Auswahl der in dieser Iteration erfüllten Anforderungen demonstriert aus Sicht des Projektteams jedoch schon den Kerngedanken des Konzepts, der im folgenden Abschnitt „Alleinstellungsmerkmale“ nochmal abschließend erläutert wird.

Zielerreichungsgrade bei der Umsetzung der Alleinstellungsmerkmale

Von den zu Beginn des Projektes formulierten Alleinstellungsmerkmalen wurden bereits nach einem ersten Feedbackgespräch einige dort formulierte Merkmale gestrichen, sodass sich ab diesem Zeitpunkt auf jene Merkmale konzentriert wurde, die das System wirklich einzigartig machen. Diese Merkmale sind im Einzelnen die Unterstützung der Tafeln durch Aggregation und Transport privater Lebensmittelspenden und die Berechnung optimaler Standorte und Zeiträume für die Sammlung solcher Spenden.

Diese Alleinstellungsmerkmale konnten im vertikalen Prototypen implementiert werden. Der Kerngedanke in erster Linie voneinander abgegrenzte Gruppierungen zur Verteilung von Lebensmitteln zu bewegen wird durch die zumindest auf Seiten des Dienstgebers implementierten Distributionsgruppen verdeutlicht.

Dem Algorithmus zur Bildung von Transportketten wurde keine so hohe Priorität eingeräumt wie der Bildung von Sammelaktionen, trotzdem wird versucht auch dieses Alleinstellungsmerkmal noch bis zum 18.01.2016 zu realisieren.

Vertikaler Prototyp

Dieser Abschnitt soll einen kurzen Überblick über die im finalen Prototypen enthaltenen Funktionalitäten verschaffen. Zur Demonstration der verteilten Anwendungslogik wurde essential use case #8 (Teilnahmevorschlag erhalten) ausgewählt.

Implementierter Kommunikationsablauf

In diesem Use case kann ein Benutzer von seinem aktuellen Standort aus seine Situation automatisiert erfassen lassen. Durch das Berechnen der Durchschnittsgeschwindigkeit in einem bestimmten Zeitintervall lässt sich das Fortbewegungsmittel erfassen. Daraus leitet sich die Transportkapazität des Users ab. Zusätzlich dazu erfasst der Android Client noch welcher Termin für den User als nächstes ansteht. Daraus ergibt sich ein Zeitfenster so wie ein Zielstandort. Dieser soll im fertigen System dazu dienen mehrere Vorschläge anhand der Route vom aktuellen Standort zum Zielstandort zu erhalten. Das Ergebnis dieser Analyse (verfügbares Zeitfenster, Fortbewegungsmittel, Standort, Zielstandort, Transportkapazität, Reichweite) wird an den Dienstgeber mittels PUT (Endpoint Benutzer/Situation) übertragen. Umgesetzt wurde die Erstellung von Sammelaktionsvorschlägen, welche am Endpunkt „Benutzer/Situation/Sammelaktionen“ mit einer GET-Operation abgefragt werden können.

Der Dienstgeber setzt den Standort der übertragenen Situation in eine Adressdarstellung um und ermittelt so den Postleitzahlbereich, in der der Benutzer sich befindet. Mithilfe dieses Ergebnisses wird aus dem Datenbestand (Tafelverein-Ressource) jener Tafelverein ermittelt, der für Spenden aus diesem PLZ-Bereich zuständig ist. So wird der Gebietsschutz der Tafeln beim generieren der Vorschläge berücksichtigt. Im Anschluss daran werden alle Angebote in diesem PLZ Bereich gefunden und eine grobe Vorsortierung anhand ihrer Abholtermine vorgenommen. Angebote deren Abholtermine vollständig außerhalb des verfügbaren Zeitfensters liegen werden nicht berücksichtigt.

Um nun eine mögliche Abholroute unter Berücksichtigung von Abholterminen, Reisedauer zu zwischen den Standorten und weiterer aus der Situationsanalyse bekannter Parameter berücksichtigen zu können wurde eine modifizierte Version des Dijkstra Algorithmus implementiert. Die Reisedauer zwischen allen potentiell in Frage kommenden Angeboten wird durch eine Anfrage an OSRM (Open Source Routing Machine) ermittelt. Das Ergebnis dieser Anfrage ist eine Distanzmatrix. Nun wird vom Standort des Benutzers aus sukzessive der Knoten mit der kürzesten Distanz zum letzten Knoten ermittelt und eine voraussichtliche Ankunftszeit berechnet. Diese Ankunftszeit wird mit den Abholterminen des betroffenen Angebots abgeglichen. Wurden alle Knoten betrachtet antwortet der Dienstgeber mit einer Sammelaktionsrepräsentation.

Der Android Client nutzt nun OSMdroid zur Einbindung einer Openstreetmap-MapView und stellt mögliche Sammelaktionen und die genauen Standorte der Angebote an der Benutzungsschnittstelle dar.

Ein Vorschlag kann dort akzeptiert werden, was im Erfolgsfall eine POST-Operation mit einer der gelieferten Repräsentationen auslöst.

Dieses POST (Endpoint /Sammelaktionen) blockiert alle in dieser Aktion angefahrenen Angebote für andere Vorschläge. Zudem wird diese Zustandsänderung mittels RabbitMQ an das Topic der Postleitzahl (/<PLZ>) gepublished. Benutzer mit Wohnort in dieser PLZ können so asynchron über das Auftreten einer Sammelaktion in Kenntnis gesetzt werden.

Zu Demonstrationszwecken wurde eine zweite Node Instanz, die den Tafel-Dienstnutzer (siehe Architekturdiagramm) darstellt implementiert. Die Sammelaktion könnte von dieser Instanz dahingehend verarbeitet werden, dass mögliche Anfahrten zu Tafel XY abgegriffen und als „Anfahrtenmonitor“ (Ursprünglich geplanter Teil der Anwendungslogik) dargestellt werden.

Wird eine Sammelaktion abgeschlossen (PUT auf die betroffene Sammelaktion) werden alle betroffenen Angebote als „abgeholt“ markiert.

Aus zeitlichen Gründen ausgelassen

Im zeitlich knappen Rahmen der Veranstaltung mussten an einigen Stellen Abstriche gemacht werden um zumindest die Kernidee des Systems zeigen zu können. Nicht mehr enthalten sind daher folgende Funktionalitäten :

- Anwendungslogik des Tafel-Dienstnutzers
- Unterschiedliche Formen des Nachrichtenaustauschs über RabbitMQ , im Prototypen wird lediglich beim POST einer Sammelaktion ein standortbezogener Broadcast abgesetzt , der alle Subscriber eines Postleitzahlbereichs benachrichtigt.
- Die UI konnte nicht in dem Umfang implementiert werden wie in den Entwürfen vorgesehen
- Transportaktionen, in denen der Transport gebündelter Spenden von Privatpersonen oder der Tafel organisiert werden sollte konnten nicht mehr implementiert werden. Die Logik zur Berechnung Distanzen anhand von Geodaten (siehe dienstgeber/lib/geo) wurde noch integriert, allerdings nicht mehr genutzt.

Ausblick

Die Implementierten Softwarekomponenten bilden eine gute Basis für zukünftige Arbeit an diesem Projekt. Die Nutzung quelloffener Softwarelösungen(RabbitMQ,OSRM) und der Verzicht auf externe Dienste (mit Ausnahme des Geocoder-Moduls, welches die Google Maps Geocoding API verwendet, aber leicht ausgetauscht werden kann) bedeutete zwar einen Mehraufwand in der Implementierung, reduziert bzw eliminiert aber die Abhängigkeit von externen Anbietern und macht das System dadurch zukunftssicherer. Die Voraussetzungen zur Implementierung der im ursprünglichen Konzept entworfenen Anwendungslogik auf Seiten des Tafel-Dienstnutzers (Abgleich von Standorten gesammelter Spenden mit der bereits vorhandenen Transportroute der

Tafel) sind bereits gegeben und erste Erfahrung in der Implementierung der Präsentationslogik mithilfe von OpenStreetMap-Tiles schafft gute Voraussetzungen für eine Weiterentwicklung. Weitere Iterationen der konzipierten U könnten nach Durchführung geplanter Evaluationen und mehr Zeit zur Einarbeitung in den Android-Styleguide noch in ihrer Qualität gesteigert werden. Aus Sicht des Projektteams konnte mit dem Implementierten vertikalen Prototypen eine gute Ausgangslage zur Implementierung eines horizontalen Prototypen geschaffen werden.