Slide 1

02.06.2019

Food in Progress

Entwicklungsprojekt interaktive Systeme – SS19

Audit 2 - Modellierung

Dozenten: Kristian Fischer, Gerhard Hartmann
Betreuer: Corinna Sofie Klein, Robert Gabriel
Studierende: Kay Ruck, Tristan Schmele

Technology
Arts Sciences
TH Köln

Zielhierarchie: https://github.com/tschmele/EISSS19RuckSchmele/wiki/Zielhierarchie

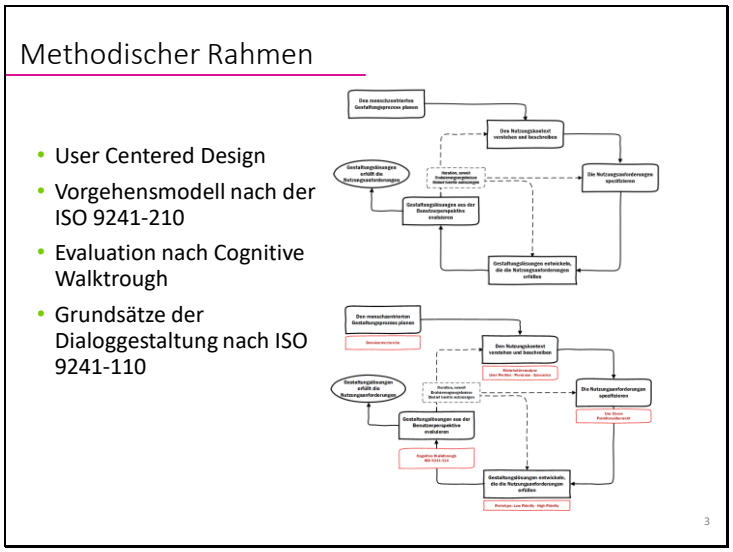
Risikoanalyse: https://github.com/tschmele/EISSS19RuckSchmele/wiki/Risiken

Slide 2

Inhalt

- Methodischer Rahmen
- Benutzermodellierung
- Aufgabenmodellierung
- Evaluationsergebnisse
- Komponentendiagramm
- Kommunikation
- Datenmodellierung

Slide 3



Vorgehensmodell

In den vergangenen Artefakten wurde festgelegt, dass das zu erstellende System den zukünftigen Benutzer beim Erreichen seiner Ziele unterstützen soll. Aus diesem Grund können Vorgehensmodelle, welche einen anwendungszentrierten Ansatz verfolgen ausgeschlossen werden. Des Weiteren sollen hohe Einarbeitungszeiten und Mehraufwand vermieden werden, damit die Nutzer in ihren unterschiedlichen Rollen schnell wechseln können.
Fazit: Damit die Ziele des Projektes bzw. des Systems erreicht werden können, sollte der Schwerpunkt auf dem Nutzer des Systems, anstatt ausschließlich auf den Funktionalitäten liegen.

Usability Engineering Techniken

Nachdem festgelegt wurde, welches Vorgehensmodell verwendet wird, musste nun eine Technik für die Umsetzung gewählt werden. Im Vorfeld wurden Kriterien festgelegt, welche die Technik erfüllen soll.
Die Technik soll…

- projektbegleitend zum Projekt sein
- flexible auf das Projekt angepasst werden können
- im zeitlichem und personellen Projektrahmen liegen
- eine gute Dokumentation besitzen
- bestenfalls ein standardisiertes Rahmenwerk sein

Aus diesen Gründen und dem Vorteil, dass nach jeder Prototype Entwicklung eine Iteration durchgeführt werden kann, wurde die Norm ISO 9241-210 gewählt. Damit das sehr allgemein gehaltene Vorgehensmodell auf das Projekt angepasst werden kann, wurde ein erweitertes

Vorgehensmodell erstellt.

Cognitive Walkthrough

Damit ein effizientes evaluieren während der Entwicklung gewährleistet werden kann, wurde sich für eine formative Technik entschieden. Des Weiteren sollte der Aufwand im Rahmen des Projektes liegen, was bedeutet, dass die Technik u.a. ermöglichen soll Evaluationen ohne Probanden durchführen zu können. Aus diesen Gründen wurde der Cognitive Walkthrough gewählt. Da der CW den Nachteil bringt das „nur“ Experten und kein wirklicher Benutzer das System testet wird nur die zweite Evaluation damit stattfinden. Die Finale Evaluation wird mit Projekt unabhängigen Probanden durchgeführt.

ISO 9241-110
Für die Erstellung des HF Prototyps war es wichtig die Grundsätze der Dialoggestaltung stark mit einzubeziehen und somit die User Experience signifikant zu steigern

Quelle:
ISO 9241-210 Vorgehensmodell: https://www.iso.org/standard/52075.html
ISO 9241-110 Grundsätze der Dialoggestaltung: https://www.iso.org/standard/38009.html

Slide 4

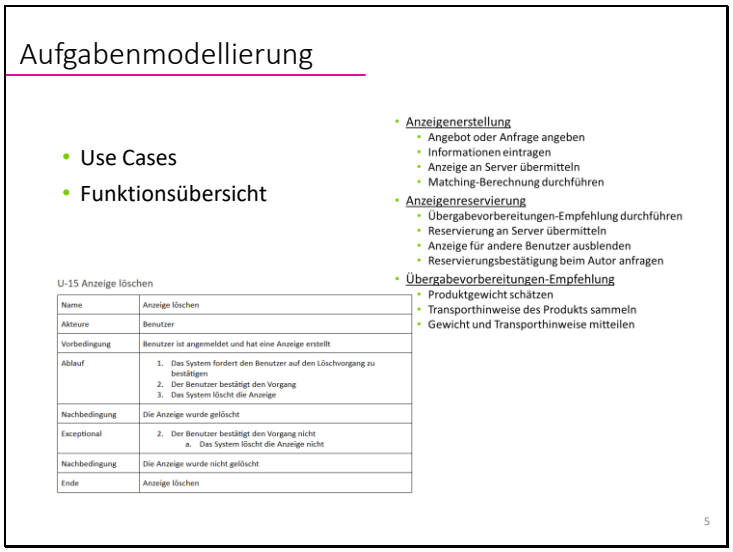
Damit die Benutzer sauber modelliert werden konnten, mussten zuerst mit Hilfe der Domänenrecherche und der Stakeholderanalyse User Profiles erstellt werden. Diese sollen den zukünftigen Nutzer auf wenige Attribute abstrahieren. Anschließend werden auf Basis der User Profiles verschiedene Personas erstellt. Zuletzt werden für jedes Persona mindestens ein Nutzerszenario erstellt.

User Profile:
Um die Benutzer des Systems noch genauer erfassen zu können, wurden erstmal mit Hilfe der Stakeholder und der Domänenrecherche User Profiles erstellt. Diese bilden einen abstrahierten Nutzer des Systems ab und werden im Laufe der Benutzermodellierung iterativ bearbeitet.

Personas:
Aus den User Profiles werden die Personas gewonnen. Diese sind mit u.a. sozio-demographischen Informationen wie Geschlecht, Alter, Berufliche Situation und Familienstand, sowie einer kurzen Beschreibung zur Person angereichert. Ziel ist es hierbei fiktive Charaktere so zu konzipieren, dass sie ausdrucksstark nachempfunden werden können.
Anmerkung: Aus Datenschutz technischen Gründen wurde auf ein Bild verzichtet.

Szenarien:
Die Szenarien verdeutlichen wie die Anwendung von dem Nutzer benutzt wird und unterstützet die Einbindung des Benutzers in den Entwicklungsprozess. Anders als bei der eher technischen Formulierung der späteren Use Cases, wurde hier vor allem weiter auf eine freiere Formulierung des Szenariotextes geachtet. So ähneln die Szenarien eher einer Geschichte mit bzw. über die Persona, als eines technischen Ablaufes.

Slide 5



Bei der Aufgabenmodellierung wird beschrieben wie das System vom Benutzer verwendet werden soll. Als erstes wurden Use Cases erstellt, diese bilden detailliert einen Anwendungsfall ab. Anschließend wurde eine Funktionsspezifikation durchgeführt, bei welcher nochmal eine Übersicht über die einzelnen Funktionalitäten erstellt wird.

Use Cases
Mit Hilfe der Szenarien konnten Use Cases erstellt werden. Im Gegensatz zu den Szenarien sollen Use Cases den technischen Ablauf zwischen den Benutzer und dem System verdeutlichen. Hierbei muss vor allem berücksichtigt werden, für welche Funktionalitäten ein Benutzer verifiziert sein muss und welche dies nicht benötigen.

Gesamtes Dokument mit allen Use Cases
https://github.com/tschmele/EISSS19RuckSchmele/blob/master/Artefakte/RuckSchmele_UseCasesSS19.pdf

Funktionsübersicht
Um das Design des User Interface mit allen Funktionalitäten einfacher gestalten zu können, wurde eine Liste aller Funktionalitäten angefertigt. Diese resultiert aus den bereits erstellen Use Cases und Anforderungen.

Anzeigenerstellung
- Angebot oder Anfrage angeben
- Informationen eintragen
- Anzeige an Server übermitteln
- Matching-Berechnung durchführen

Anzeigenabbruch
- Anzeige löschen
- Alle anzeigenbezogenen Daten löschen

Anzeigenreservierung
- Übergabevorbereitungen-Empfehlung durchführen
- Reservierung an Server übermitteln
- Anzeige für andere Benutzer ausblenden
- Reservierungsbestätigung beim Autor anfragen

Reservierungsbestätigung
- Akzeptieren der Reservierung
- Freigabe der Kontaktadressen

Benutzerbewertung
- Auswahl einer Bewertungsstufe
- Mögliche Angabe eines Kommentars
- Bewertung an Server übermitteln
- Benachrichtigung des Benutzers über die Bewertung

Matching-Berechnung
- Wahrscheinlichsten Abnehmer berechnen
- Berechnete Benutzer über Anzeige benachrichtigen

Übergabevorbereitungen-Empfehlung
- Produktgewicht schätzen
- Transporthinweise des Produkts sammeln
- Gewicht und Transporthinweise mitteilen

Übergabebestätigung Anbieter
- Erfolgreiche Abgabe eintragen
- Bestätigung an Server übermitteln
- Möglichkeit Benutzerbewertung durchzuführen

Übergabebestätigung Empfänger
- Erfolgreiche Annahme eintragen
- Bestätigung an Server übermitteln
- Möglichkeit Benutzerbewertung durchzuführen

Lebensmittel-Eintragung Manuell
- Lebensmittelinformationen eintragen
- Haltbarkeit-Schätzung durchführen
- Neues Lebensmittel an Server übermitteln
- Lebensmittelsortierung durchführen

Lebensmittel-Eintragung Scan
- Scan einer Lebensmittelladen Rechnung
- Erstellen einer Liste von Lebensmitteln
- Haltbarkeit-Schätzung durchführen
- Überprüfung der Liste durch den Benutzer
- Liste an Server Übermitteln
- Lebensmittelsortierung für Liste durchführen

Lebensmittel-Bearbeitung
- Angabe der zu aktualisierenden Informationen
- Neue Informationen an Server Übermitteln

Lebensmittelsortierung
- Ideale Lagerbedingungen heraussuchen
- Lebensmittel in bestes Lager eintragen
- Manuelle Prüfung der Sortierung durch den Benutzer

Haltbarkeit-Schätzung

- Voraussichtliches Verfallsdatum schätzen

Lebensmittel-Zustandsschätzung

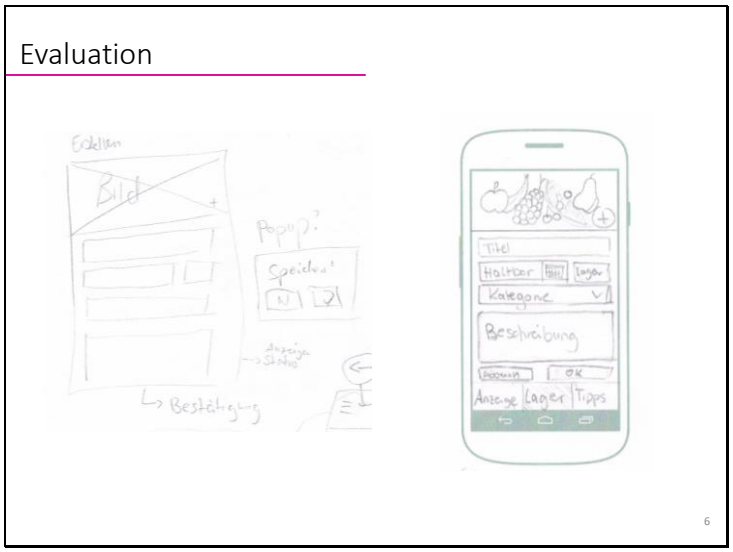
- Aktuellen Zustand der gelagerten Lebensmittel einschätzen
- Bei kritischem Zustand Benutzer darauf hinweisen

Vorratsaktualisierung

- Lebensmittel aus Lager löschen

Aktuelle Liste unter
https://github.com/tschmele/EISSS19RuckSchmele/wiki/Funktionalitäten

Slide 6



Ablauf
Zuerst wurden Skizzen auf Papier angefertigt, welche zu einem Low-Fidelity-Prototype, auch Paperbased Prototpye, wurden. Dabei kann ohne viel Aufwand grundlegende Funktionalitäten visualisiert und schnelle Änderungen im GUI Design verwirklicht werden. Danach wurde eine saubere Ausarbeitung des ersten Paperbased Prototype angefertigt. Dieser wurde auch auf Papier erstellt. Der Detailgrad ist sehr viel höher als bei den ersten Prototypen. Zudem lassen sich hier bereits Elemente des High Fidelity Prototypen erkennen. Die Erstellung des zweiten Prototypens war ein iterativer Prozess, in dem mittels Szenarien verschiedene Anwendungsfälle simuliert und mittels des Cognitive Walkthrough der Prototyp evaluiert wurden. Am Ende der Evaluation wurden alle Änderungen notiert und später im digitalen High-Fidelity-Prototype angepasst.
Zuletzt wurde ein digitaler High-Fidelity-Prototype erstellt. Dieser basiert auf den zweiten papierbasierten Prototypen und seiner Evaluation. Beide Projektparteien, Designer und Entwickler, haben anschließend den Prototype zusammen evaluiert, um zu überprüfen ob, das aktuelle Design den Anforderungen und der Funktionsübersicht entspricht. Nach kleineren Anpassungen wurde der Prototype als geeignet empfunden, um die nächste Evaluation mit, von Projekt unabhängigen, Testern durch zu führen.

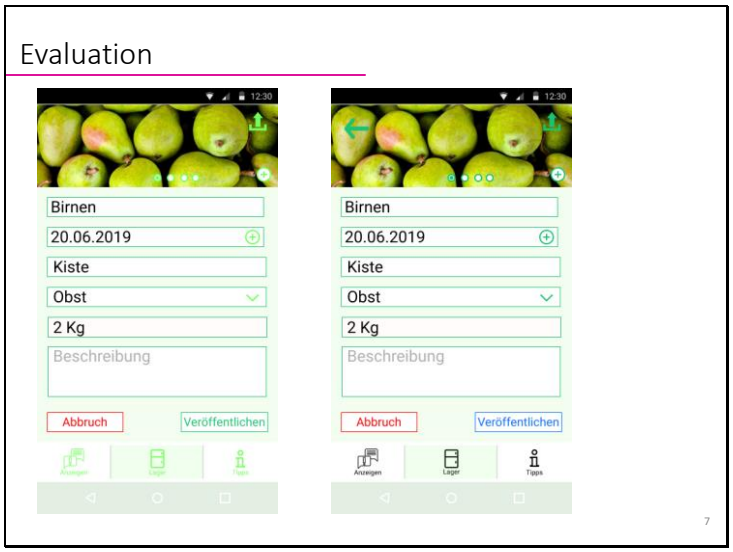
Low Fidelity Prototype
Zu Beginn wurde ein LF Prototype angefertigt, da die Erstellung schnell und unkompliziert ist. Dieser war einfarbig und wurde mit Bleistift auf Papier gezeichnet. Er diente dazu die ersten grundlegenden Funktionalitäten zu visualisieren, sowie für ein erste eigene Evaluation. Der Detailgrad war hierbei noch sehr rudimentär und einfach gehalten.

High Fidelity Prototype
Aus den Low Fidelity Prototype und seiner abschließenden Evaluation wurde der High Fidelity Prototype erstellt. Dieser bietet eine weitaus höhere Detailabbildung und diente u.a. dafür mit Probanden die Benutzeroberfläche zu evaluieren.

Der digitale Prototype wurde mittels der Software Figma erstellt und mit kleineren Interaktionen von Marvel angereichert.

Template:
https://i.pinimg.com/originals/43/8e/de/438ede24264eb9552b635d34142eac29.png

Slide 7

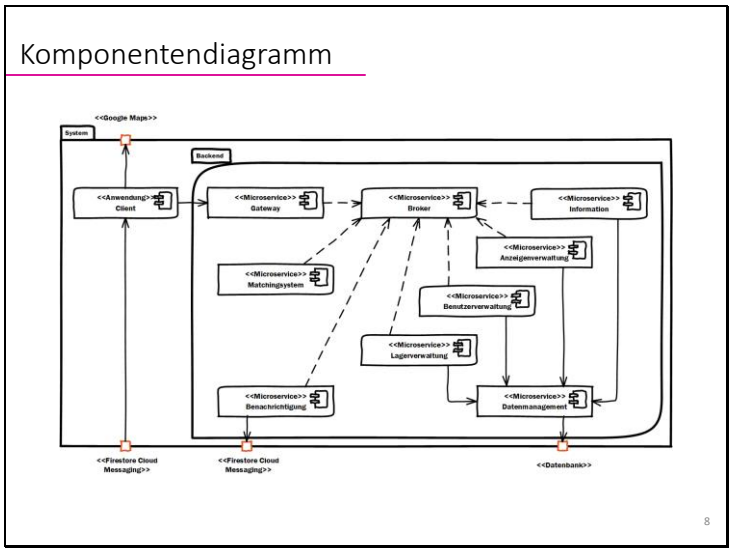


Erste Evaluation:
Die erste Evaluation wurde nach der Erstellung des zweiten papierbasierten Prototypens, mittels Cognitive Walkthroughs durchgeführt. Die Simulation fand auf Basis der angefertigten Szenarien statt. Die daraus resultierenden Ergebnisse wurden in den ersten digitalen High-Fidelity-Prototype übernommen. Im Fokus der Evaluation stand vor allem die Erreichbarkeit der Designelemente.
Erkenntnisse:
Fehlen wichtiger Navigationselemente
Durch das Fehlen wichtiger Navigationselementen wie eines „Zurück“-Pfeils, fiel die Navigation von Unterseiten zu den Hauptseiten oft sehr schwer. Zu dem konnten einige Seiten dadurch schwer oder gar nicht erreicht werden.
Fehlen einer Übersichtseite der eigenen Anzeigen
Eine Übersichtsseite zu den eigenen Anzeigen fehlte komplett und wurde noch zusätzlich erstellt. Dort soll der Benutzer seine Angefragten Lebensmittel so wie, die selbst erstellten Anzeigen sehen können.
Visualisierung der Notifications im Lock Screen
Damit es schon mal eine grobe Vorstellung gibt, wie die Notifications im Lock Screen aussehen und agieren, wurde hierfür eine zusätzliche Seite erstellt.
Hinzufügen des Lagers wurde vereinfacht
Das Hinzufügen des Lagers wurde in die Lagerübersicht selbst gelegt, so kann der Benutzer mit einer einfachen Interaktion ein neues Lager erstellen.
Hinzufügen von Lebensmittel ohne Lager
Wie bei Punkt 4 Schon genannt wurde das Lager von dem „Plus“-Icon auf die den Screen selbst gelegt. Somit konnte das Icon mit der Interaktion Lebensmittel ohne Lager belegt werden. Hierbei wird allerdings auch potential zum Missinterpretieren der Funktionalität gesehen.

Zweite Evaluation
Die zweite Evaluation fand zwischen den beiden Projektparteien statt. Hierbei wurden einzelne Designelemente beurteilt und bei Bedarf bearbeitet. Im Gegensatz zur ersten Evaluation wurde hier nicht nur Erreichbarkeit der einzelnen Screens, sondern auch auf die Selbstbeschreibungsfähigkeit der Designelemente geachtet. Des Weiten wurde erstmals die Farbgebung diskutiert. Das Ergebnis wurde als Grundlage für den zweiten digitalen High-Fidelity-Prototype hergenommen.
Erkenntnisse:
Hinzufügen einer Löschoption für Reservierungen
Das Löschen der Reservierungen ist nun mittels eines kleinen „Minus"-Icons möglich, welches auf die Reservierungsanzeige platziert wurde.
Farbgebung mit Hellgrün ist schwierig
Es wurde diskutiert, ob insgesamt die Farbgebung von hellgrün nicht als „schwierig" einzuordnen ist. Denn die Erkennbarkeit der Designelemente kann je nach Anzeige gerät sehr stark variieren. Trotzdem wurden erst einmal nur kleinere Änderungen vorgenommen.
Änderung des Logouts von hellgrün zu grau und rot
Das Logout-Symbol wurde von grün zu einem grau und rot geändert. Dies wird als verständlicher erachtet.
Änderung der Anfrage und der Bestätigung von hellgrün zu blau
Sowohl die Frage- bzw. Ausrufezeichen der Anfragen als auch die Bestätigung der Reservierung wurden zu einem blau geändert, damit sich diese besser absetzen können.

Dritte Evaluation:
Nach der zweiten Evaluation wurde der zweite digitale High-Fidelity-Prototype für ausreichend ausgereift erachtet, um mit den Tests an projektunabhängigen Personen zu beginnen. Im Folgenden wird kurz das Vorgehen während der Tests erläutert.
Vorgehen:
Zu Beginn sollte sich jeder Proband mit der Anwendung vertraut machen. Anschließend wurden ihnen eine Rolle gegeben und eine Aufgabe gestellt, welche sie bearbeiten sollten. Während sie diese Aufgaben lösten, waren die Probanden dazu angewiesen ihre Schwierigkeiten zu verbalisieren. Nach Abschluss der Aufgabe, wurde ihnen eine neue gegeben. Zum Schluss wurden die Probanden zu der Anwendung befragt. Hierbei konnten Probanden nochmals ihre Kritikpunkte zu der Anwendung selbst aufführen und im Prototyp visualisieren.
Erkenntnisse:
Farbliche Anpassung der Akzentfarbe zu einem dunkleren Grünton
Die Farbe konnte nur schlecht von dem pastellgrünen bzw. weißen Hintergrund unterschieden werden und die Lesbarkeit ist je nach Anzeige Gerät drastisch gesunken. Somit wurde nun doch der Akzentton von einem hellen Grünton zu einem gesättigten mittel Grünton, sowie einem Blauton geändert.
Änderung des Farbtons der Botton Navigation zu einem Schwarz
Sowie die Akzentfarbe wurde ebenfalls die Farbe der Botton Navigation geändert. Allerdings wurde hier ein Schwarzton gewählt.
Vereinheitlichung der Botton Navigation
Die Botton Navigation wurde so angepasst, dass diese für jeden Screen zu den gleichen Seiten führt. Diese Anpassung war wichtig, um die Konsistenz der Designelemente zu gewährleisten.
Bild: https://pixabay.com/photos/fruit-pear-pear-basket-sweet-bio-1534494/

Slide 8



Gestrichelte Pfeile: Asynchron
Durchgezogene Pfeile: Synchron

Microservices

Aktuell stellt sich die Frage ob Server wie bisher in Form von einem großen Software Monolithen oder als mehrere unabhängige kleinere Microservices umgesetzt werden sollten. Durch ihre lose Kopplung heben sich Microservices stark von den bisher üblichen Software Monolithen ab. Für die Benutzer hat dies meist keine direkten Auswirkungen, aber sowohl in der Entwicklung als auch bei der Instandhaltung und Erweiterung des Systems besitzen sie deutliche Vorteile. Jede Komponente kann unabhängig vom gesamten System implementiert, getestet und gewartet werden. Dadurch sind die Entwickler in der Umsetzung deutlich unabhängiger und einzelne Komponenten können bereits frühzeitig in Betrieb genommen und von anderen während der Entwicklung genutzt werden. Zudem können bei starker Belastung der Server anschließend mit Service Discovery in Echtzeit die Anfragen auf mehrere Instanzen der gleichen Komponenten verteilt werden, um so die Reaktionsgeschwindigkeit des Systems auf alle Anfragen deutlich zu erhöhen. Lediglich die Kommunikation der einzelnen Komponenten untereinander wird durch diesen Ansatz umständlicher, was allerdings nicht die langfristigen Vorteile überwiegt. Aus diesem Grund wird das System in diesem Projekt mit mehreren Microservices umgesetzt.

Kommunikation

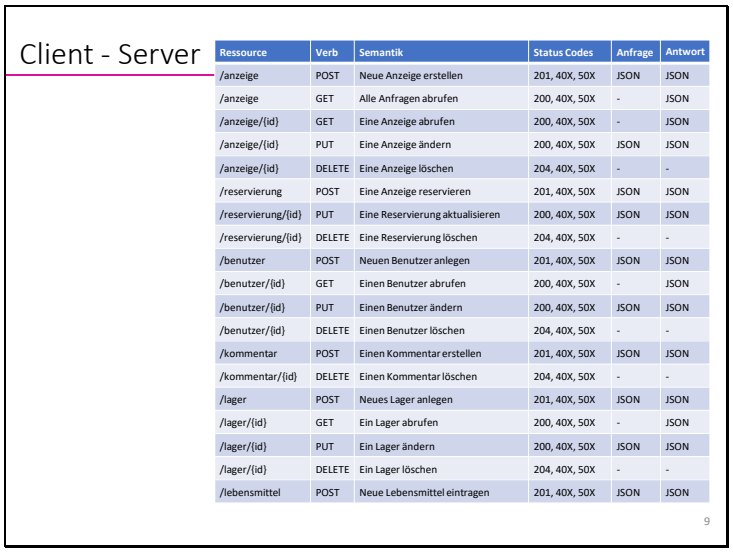
Im System muss zum einen der Client mit dem Backend und außerdem die einzelnen Microservices im Backend untereinander kommunizieren. Da der Client hauptsächlich nach Daten aus dem Server fragt und nur mit einer Antwort weiterarbeiten kann, ist es sinnvoll diese Kommunikation synchron über eine REST-Schnittstelle laufen zu lassen. Allerdings werden serverseitige Benachrichtigungen mittels Firebase Cloud Messaging asynchron an die

betroffenen Endgeräte gesendet, da in diesem Fall keine Antwort benötigt wird. Die Microservices sollen weitestgehend unabhängig voneinander funktionieren und nicht unmittelbar durch den Ausfall einer anderen Komponente beeinträchtigt werden. Um dies zu fördern, werden die einzelnen Services so implementiert, dass sie möglichst wenig von anderen Services wissen. Eine synchrone Kommunikation würde dieses Vorhaben einschränken. Aus diesem Grund werden die Microservices untereinander mittels Topic based Messaging arbeiten. Dadurch können sie nahezu vollständig unabhängig arbeiten und müssen lediglich wissen wohin sie ihre Ergebnisse publishen müssen und können durch subscriptions auf Informationen waren, ohne zu wissen woher diese kommen.

Externe Dienste

Um einen besseren Einfluss auf den tatsächlichen Datenschutz der Benutzer zu haben, sollen im finalen Produkt eigene Server, Datenbanken und Kartendienste genutzt werden. Dafür sind Open Source Dienste wie OpenStreetMaps und MongoDB geplant.
Um jedoch frühzeitig in der Entwicklung bereits einen funktionierenden Prototypen bereit stellen zu können, werden in den Anfangsphasen der Entwicklung Google Dienste wie Firebase und Google Maps genutzt. Diese sind in einem limitierten Umfang kostenlos nutzbar und können mit vergleichsweise wenig Aufwand eingesetzt werden. Sollte der Prototyp gut ankommen und alle Funktionalitäten in gewünschtem Umfang funktionieren, werden dieses Services mit eigenen Instanzen von bereits genannten Open Source alternativen ersetzt.

Slide 9



Client - Server

Ressource	Verb	Semantik	Status Codes	Anfrage	Antwort
/anzeige	POST	Neue Anzeige erstellen	201, 40X, 50X	JSON	JSON
/anzeige	GET	Alle Anfragen abrufen	200, 40X, 50X	-	JSON
/anzeige/{id}	GET	Eine Anzeige abrufen	200, 40X, 50X	-	JSON
/anzeige/{id}	PUT	Eine Anzeige ändern	200, 40X, 50X	JSON	JSON
/anzeige/{id}	DELETE	Eine Anzeige löschen	204, 40X, 50X	-	-
/reservierung	POST	Eine Anzeige reservieren	201, 40X, 50X	JSON	JSON
/reservierung/{id}	PUT	Eine Reservierung aktualisieren	200, 40X, 50X	JSON	JSON
/reservierung/{id}	DELETE	Eine Reservierung löschen	204, 40X, 50X	-	-
/benutzer	POST	Neuen Benutzer anlegen	201, 40X, 50X	JSON	JSON
/benutzer/{id}	GET	Einen Benutzer abrufen	200, 40X, 50X	-	JSON
/benutzer/{id}	PUT	Einen Benutzer ändern	200, 40X, 50X	JSON	JSON
/benutzer/{id}	DELETE	Einen Benutzer löschen	204, 40X, 50X	-	-
/kommentar	POST	Einen Kommentar erstellen	201, 40X, 50X	JSON	JSON
/kommentar/{id}	DELETE	Einen Kommentar löschen	204, 40X, 50X	-	-
/lager	POST	Neues Lager anlegen	201, 40X, 50X	JSON	JSON
/lager/{id}	GET	Ein Lager abrufen	200, 40X, 50X	-	JSON
/lager/{id}	PUT	Ein Lager ändern	200, 40X, 50X	JSON	JSON
/lager/{id}	DELETE	Ein Lager löschen	204, 40X, 50X	-	-
/lebensmittel	POST	Neue Lebensmittel eintragen	201, 40X, 50X	JSON	JSON

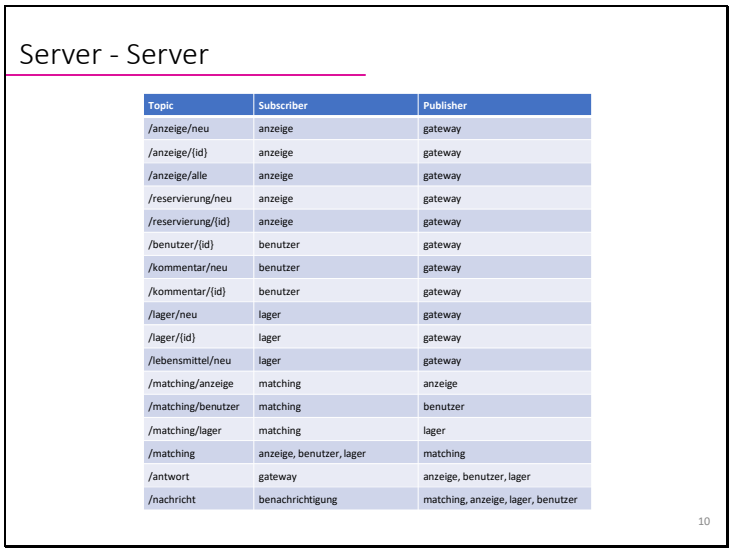
9

200 – ok
Anfrage wurde erfolgreich bearbeitet
201 – created
Ressource wurde erfolgreich erstellt
204 – no content
Anfrage wurde erfolgreich bearbeitet, aber Antwort wurde bewusst ohne daten gesendet
400 – bad request
Anfrage war fehlerhaft
401 – unauthorized
Anfrage kann nicht ohne gültige Authentifizierung bearbeitet werden
404 – not found
Angeforderte Ressource kann nicht gefunden werden
408 – request time out
Anfrage konnte innerhalb der festgelegten Zeitspanne nicht vollständig empfangen werden
409 – conflict
Anfrage wurde unter falschen Annahmen gestellt
410 – gone
Angefragte Ressource wurde dauerhaft entfernt
500 – internal server error
Nicht spezifizierter, unerwarteter Serverfehler
501 – not implemented
Funktionalität, um Anfrage zu bearbeiten, wird nicht bereitgestellt
502 – bad gateway
Server konnte seine Aufgabe nicht erfüllen, da er seinerseits eine ungültige Antwort erhalten hat
504 – gateway timeout
Server konnte seine Aufgabe nicht erfüllen, da er von seinerseits genutzten Diensten keine Antwort innerhalb einer festgelegten Zeitspanne erhalten hat

JSON

Die übertragenen Daten sollen in einem einheitlichen Datenformat verpackt werden. Die beiden populärsten Optionen sind XML und JSON. Durch die Möglichkeit XML Dateien validieren zu lassen, um fehlerhafte Übertragungen zu vermeiden, ist XML eigentlich besser geeignet. Allerdings sind XMl Dateien in der Regel größer als JSON Dateien. Da PlanA die Benutzer in allen Situationen gut unterstützen zu können sollte die Datennutzung der Anwendung so gering wie möglich gehalten werden. Zudem bekommt das System von fast allen bisher geplanten APIs und externen Servicen Antworten im JSON Format. Daher bietet es sich an interne Kommunikation im gleichen Format durchzuführen.

Slide 10



Server - Server

Topic	Subscriber	Publisher
/anzeige/neu	anzeige	gateway
/anzeige/{id}	anzeige	gateway
/anzeige/alle	anzeige	gateway
/reservierung/neu	anzeige	gateway
/reservierung/{id}	anzeige	gateway
/benutzer/{id}	benutzer	gateway
/kommentar/neu	benutzer	gateway
/kommentar/{id}	benutzer	gateway
/lager/neu	lager	gateway
/lager/{id}	lager	gateway
/lebensmittel/neu	lager	gateway
/matching/anzeige	matching	anzeige
/matching/benutzer	matching	benutzer
/matching/lager	matching	lager
/matching	anzeige, benutzer, lager	matching
/antwort	gateway	anzeige, benutzer, lager
/nachricht	benachrichtigung	matching, anzeige, lager, benutzer

10

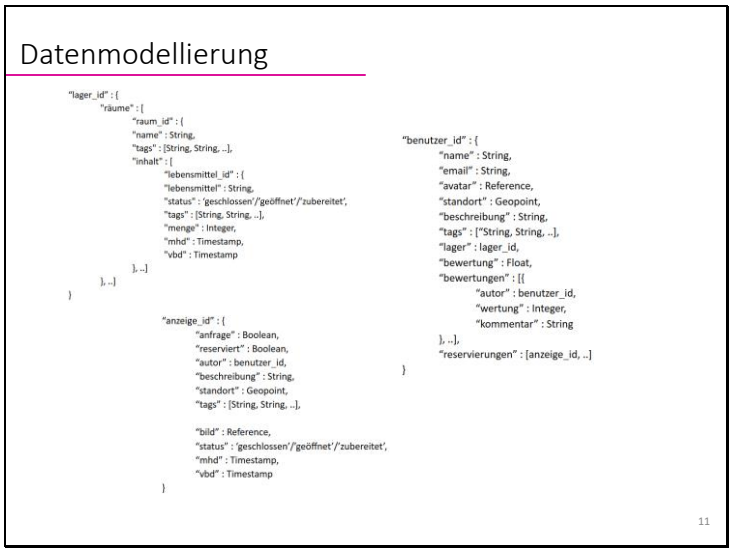
/anzeige - /lebensmittel
→ Weiterleitung der REST Anfragen des Clients durch die Gateway Komponente

/matching
→ Kommunikation, welche zur Berechnung der wahrscheinlichen Abnehmer benötigt wird

/antwort
→ Rückmeldungen auf die weitergeleiteten Anfragen des Clients

/nachricht
→ Ereignisse bei denen Benutzer von Server benachrichtigt werden müssen

Slide 11



Die Daten werden in einer NoSQL Datenbank gespeichert und dafür in Sammlungen und Dokumente sortiert.

Jeweils Lager, Benutzer und Anzeigen werden einer gleichnamigen Sammlung gespeichert. Die Strukturen der darin enthaltenen Dokumente sind auf der Folie allgemein zu lesen. „lager_id“, „benutzer_id“ und „anzeige_id“ beziehen sich auf die jeweiligen IDs der Dokumente.
Geopoints bestehen aus Längen- und Breitengrad.
„mhd“ und „vbd“ enthalten je nach Lebensmittel das Mindesthaltbarkeitsdatum oder das Verbrauchsdatum.
„tags“ enthalten ein Array mit einer Auswahl an vom System vorgegebenen Strings, welche an verschiedenen Stellen im System (zB Filtern von Anzeigen) genutzt werden können.

Anzeigen können entweder Anfragen oder Angebote sein. Bei Anfragen werden die unteren 4 Felder nicht benötigt.