

# Iteration

## Datenbank:

Bisher wurde MongoDB intern genutzt, was aber im Rahmen unseres Projekts zu Problemen führen könnte, da Außenstehende wie Mentoren und Dozenten, nicht zu jeder Zeit auf die interne Datenbank zugreifen können. Des Weiteren ist es allgemein von Vorteil eine externe Datenbank in Form einer Cloud zu nutzen, da so einfacher und verlässlicher auf die gespeicherten Daten zugegriffen werden kann. Wir haben uns im Rahmen dieses Projekts für die kostenlose Version der MongoDB Atlas Cloud entschieden, da vorher schon MongoDB genutzt wurde war das umrüsten und transferieren der Daten auf die Cloud schneller und einfacher. Uns ist bewusst das Daten durch ein Backup gesichert werden sollten, in unserem Fall verzichten wir auf ein Datenbank Backup, da es sich nur um Testdaten handelt.

Im 3. Audit wurde uns nahegelegt, dass unser Projekt eine große Lücke im Bereich der Computer-Mensch-Interaktion aufweist und diese dringends gefüllt werden sollte, um das Modell Entwicklung Interaktiver System erfolgreich zu bestehen. Daher haben wir uns dazu entschieden nochmals viel Artefakte zu überarbeiten bzw. zu iterieren.

Des Weiteren soll sich über das Gerät selbst identifiziert werden und um Spenden den richtige Organisationen zuzuweisen, soll eine Art Katalog vom Server erfragt werden, welcher clientseitig mit der Verkaufshistorie verglichen wird, um so zu berechnen, welche Spendenorganisation für den Nutzer die Richtige ist. Unser Projekt beinhaltet nur den Vorgang der Sortierung der Nachhaltigkeit gekaufter Produkte bis zur Organisations Empfehlung, dass heißt Vorgang der wirklichen Spenden an die Organisation wird nicht gezeigt bzw. lässt das Rahmen unseres Projekts nicht zu.

Es wurden neue Internetrecherchen durchgeführt bei denen das Hauptaugenmerk das plastikfreie Einkaufen war, da die Frage des öfteren zu Tage kam, wie Nutzer, welche ihre eigenen Behälter mitbringen in unser System integriert werden. Dabei bin ich zu dem Entschluss gekommen, dass es sehr schwer ist Nutzer, welche keine vom Supermarkt gestellten Verpackungen nutzen, in mein System zu integrieren, da es auf Barcodes ausgelegt ist und es sehr schwer ist in allgemeinen Supermärkten, wie beispielsweise Edeka, Penny-Markt, etc.mit den vom Markt gestellten Codes auszukommen, wie für Obst oder Müsli (also Produkte die dort in eigenen Behälter transportiert werden können, also ohne Verpackung) da diese Codes vom Markt selbst generiert werden und sich des weiteren auch von Markt zu Markt unterscheiden, sie werden also für mein System, welches im Umfang dieses Projekts entwickelt wird nicht berücksichtigt. Trotzdem hätten Nutzer mit eigenen Behälter integriert werden können, indem auf dem Kassenbon eine Kennzeichnung des verpackungsfreien Produkts gemacht wird und so beispielsweise ohne wirkliche Informationen über das Produkt, die Summe der Spende durch das Kaufen des Produkts erhöht wird, sozusagen als größere Belohnung. Dadurch sollte der Nutzer noch mehr in Richtung Änderung des Einkaufsverhaltens gedrängt werden ("behavior change").

Auch Modelle wie das Architektordiagramm und REST-Tabelle wurden iteriert, dazu mehr auf den anderen Folien.

# Methodischer Rahmen

Zunächst wurde ein Design-Prinzip ausgewählt, auf dessen Grundlage ein für das Projekt geeignetes Vorgehensmodell aus der MCI gewählt wurde.

## Design-Prinzipien:

Es wird zwischen dem benutzerzentrierten Ansatz "user centered design" und dem aufgaben- und zielorientierten Ansatz "usage centered design" unterschieden. Für das Projekt wird das Design-Prinzip "usage centered design" verwendet, da hier die Merkmale der Benutzung Ausgangspunkt für die Modellierung des Systems sein sollen. Im Rahmen des Projekts ist dieser Ansatz meiner Meinung nach der bessere.

## Usage-Centered-Design (Lockwood und Constantine)

- User Roles und Role Map => Role Model
- Task Cases und Task Map => Task Model
- Abstract Prototypes und Navigation Map => Content Model
- Visual and Interaction Design

Auf Personas wurde in meinem Projektkontext verzichtet, da laut Constantine und Lockwood, nur Modelle modelliert werden sollten die auch effizient seien. Meiner Meinung nach kann jeder die Applikation nutzen der Einkäufe tätigt. Es ist irrelevant, welcher gesellschaftlichen Klassen man zugeordnet ist, da nur der Besitz eines Handys eine Rolle spielt und die Tätigkeit Lebensmittel zu erwerben, deswegen macht es keine Sinn sich in verschiedene Personen hineinzusetzen (durch Personas).

## Actors

A1: Mitarbeiter einer Spendenorganisation

A2: Person die Spenden bezieht(Bsp. Landwirtschaft)

A3: Supermarktbesucher

A4: Supermarktmitarbeiter

Es wurden vorerst alle "Spieler" des "Spielfeldes" modelliert. Diese müssen laut Constantine und Lockwood in zwei Klassen unterteilt werden. Direkte "Spieler" und indirekte "Spieler". Spendenorganisationen und Personen die Spenden der Organisationen beziehen zählen zu den indirekten "Actors", da sie keinen wirklichen Einfluss auf das System haben. Aber Supermarktbesucher und Supermarktmitarbeiter, welche die App nutzen zählen zu direkten Nutzern des Systems.

# User Roles

R1: Scannen der Codes

R2: Identifizierung des Käufers

R3: Kauf nachhaltiger Produkte

R4: Wahl der Spendenorganisation

R1: Scannen des Codes

Context: Steht meistens unter Druck, da er beim Scannen der Produkte eine gewisse Geschwindigkeit aufbringen muss/soll, um möglichst lange Schlangen an der Kasse zu verhindern. Dabei immer Blick auf Schlange, steht unter Druck.

Charakteristika: Gleicher Vorgang, immer und immer wieder, könnte unter Druck beim Scannen zu Fehlern neigen.

Kriterien: Sollte möglichst schnell arbeiten, bestimmte Flexibilität aufweisen

R2: Identifizierung des Käufers

Context: Ist eine Spezialisierung von R1. Schwer für Mitarbeiter immer von nicht App Nutzern zu unterscheiden, da es durchaus vorkommen kann das Nutzer der App selbständig sagen, dass sie die App nutzen.

Charakteristika: Gleicher Vorgang sehr trivial zu R1

Kriterien: Niemals vergessen QR-Code zu scannen, Aufmerksamkeit, heißt fragen ob bestimmte Kooperationen genutzt werden, wie beispielsweise Payback, (die entwickelte App), etc

R3: Kauf nachhaltiger Produkte

Context: Käufer versucht möglichst nachhaltig einzukaufen, kann aber nicht immer auf alle Einflüsse bzw. Herkunft, Produktionsort, etc achten.

Charakteristika: Meistens sind die Einkäufe bzw. Supermarkt Besuche sehr trivial aber es kann durchaus vorkommen das ein Einkauf unter Stress oder einer bestimmten Schnelligkeit getätigt werden muss.

Kriterien: Sollte umweltbewusst einkaufen, es sollte auf Herkunft des Produkts geachtet werden, bestimmte Siegel sollen beachtet werden, eventuell wenn möglich auf vegane Ernährung, eigene Behälter sollten mitgebracht werden, möglichst wenig Verpackungsrest sollte erworbt werden.

R4: Wahl der Spendenorganisation

Context: Der Spender/Käufer kann selbständig entscheiden wohin letztendlich seine prozentuale Spende ankommt, auch unabhängig von getätigten Einkäufen.

Charakteristika: Sollte immer aufmerksam, nach getätigten Einkäufen verfolgen wohin seine Spende gelangt. Sollte sich bevor eigener Spendenwahl überlegen, ob wirklich der Spendenwunsch seinen Einkaufsverhalten entspricht, da Spenden nach Einkaufsverhalten berechnet werden.

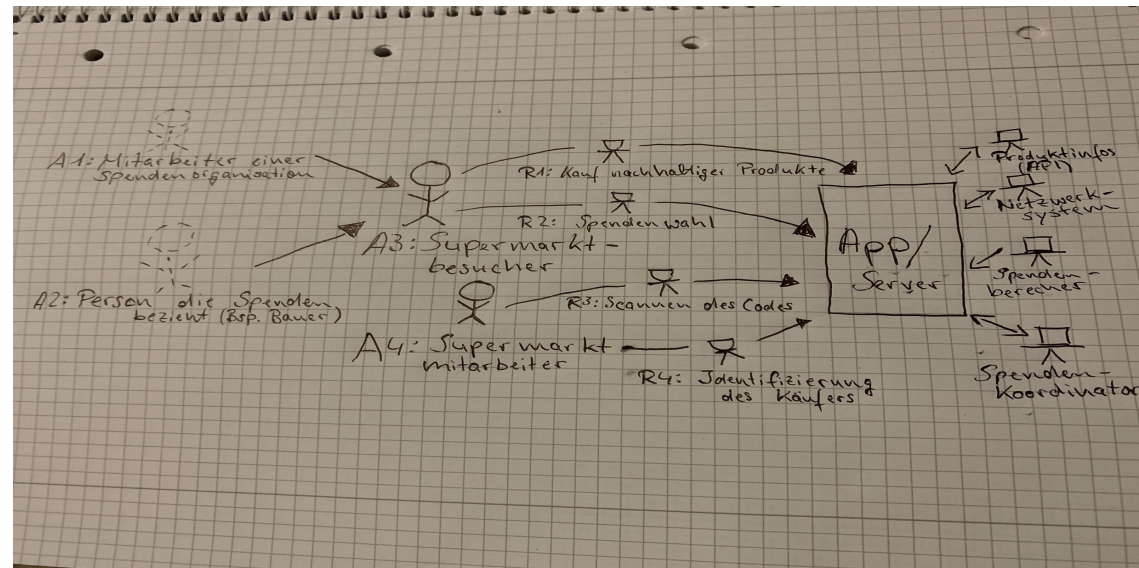
Kriterien: Sollte ein gewisse Ziel haben, bzw. wissen sollen was verbessert werden soll, sollte nicht zufällig wählen, sollte sich informieren.

User Roles stellen nicht wie beispielsweise Personas reale Personen da, sie sind eher technische und strukturierte Modelltypen. Eine User Rolle wird durch Verhalten, Erwartung und Interesse zu einem System definiert. Nach Constantine sei der einfachste Weg, eine Aktivität zu modellieren, erstmal einen Plan des „Spielfelds“ mit allen Teilnehmern anzufertigen. Es sollen also alle Spieler auf "dem Feld" erfasst werden, auch genannt Actors. Diese Actors müssen zwischen User Actors und System Actors unterschieden werden. Des Weiteren werden User Actors noch in direkten User Actors und indirekten User Actors unterschieden. Nach Constantine und Lockwood sollen

User Roles nach einer kartenbasiert Technik modelliert werden (base-card technique).

Diese Karten bestehen aus einem Context, Characteristics und bestimmten Kriterien.

## Context Map



# Essential Use Cases

Benutzer Aktion	System Reaktion
1. User erfragt bereits getätigte Einkäufe oder lässt Spendenorganisation direkt berechnen	
	2. System erkennt User und berechnet anhand getätigter Einkäufe, die zu den Einkäufen passende Organisation
3. User ist zufrieden, mit berechneter Organisation, falls nicht wird die Option selbst wählen genutzt	
	4. System zeigt möglich Organisationen an
5. User wählt gewünschte Organisation	
	6. Ändert Spendenorganisation des Users, zu gewünschter Organisation
	7. Spende wird zukünftig an gewählte Organisation gespendet

Kunde nutzt Spendenberechnung und Spendenorganisation selbst wählen.

# Essential Use Cases

Benutzer Aktion	System Reaktion
1. Kunde sagt das er Nutzer der App ist (vergleichbarer Vorgang Payback)	
	2. Mitarbeiterin fordert Identifikation
3. Kunde öffnet App und zeigt QR-Code	
	4. Supermarktmitarbeiterin scannt Code
	5. Mitarbeiterin scannt Produkte(irrelevant, ob User sich zuerst identifiziert oder erst Produkte gescannt werden, dass System erkennt den Unterschied zwischen UserId und Barcodes. Theoretisch auch Produktscan, dann Identifikation und dann Fortsetzung des Produktscans möglich.)
	6. System erstellt Nutzer und fügt ersten Kassenbon hinzu. Zum Kassenbon werden, durch das System nur nachhaltige Produkte hinzugefügt

Kunde nutzt App das erste Mal an der Kasse (Spendenorganisation wird automatisch eingetragen, da noch keine Einkäufe getätigt wurden und so keine Berechnung anhand der Einkäufe geschehen kann)

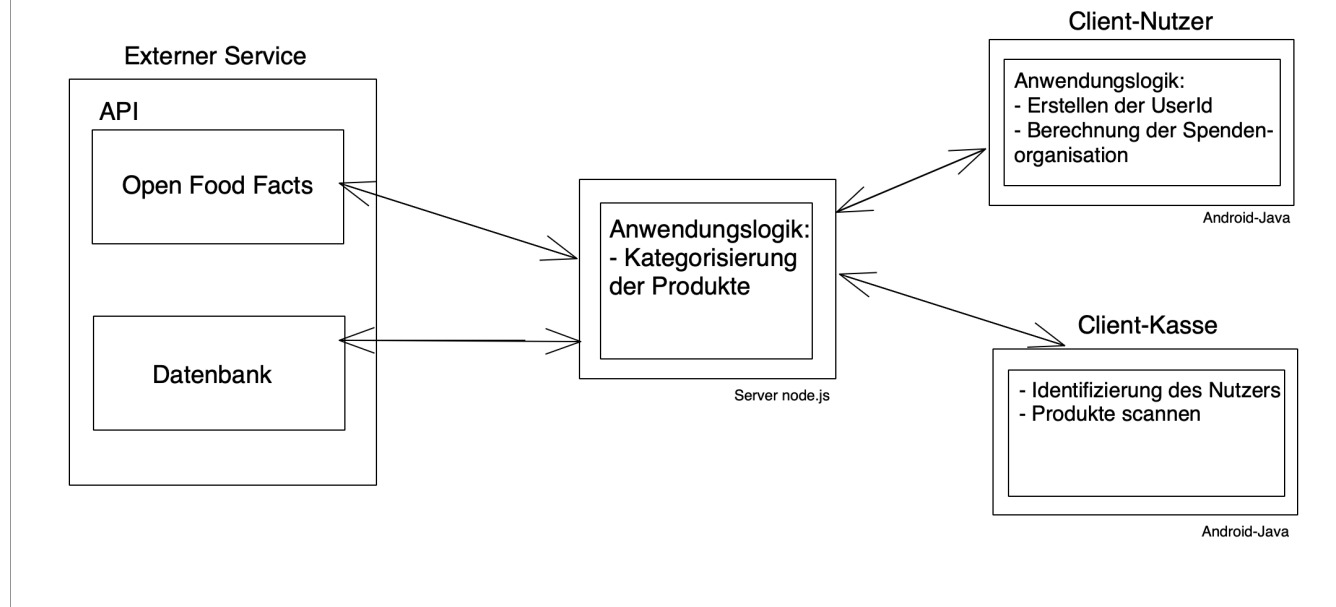


# Essential Use Cases

Benutzer Aktion	System Reaktion
1. Fordert Spendenübersicht an	
	2. Identifiziert User anhand der userId
	3. Zeigt alle bisherigen Einkäufe an, mit Spenden etc.
4. Schaut sich Übersicht an	

Spendenübersicht ansehen

# Architekturdiagramm



## Datenaustausch:

Durch ein GET-Request zur API werden Produktinfos in Form einer JSON-Objekts abgerufen.

Auf die Datenbank kann durch GET, PUT, POST, DELETE zugegriffen werden(alles genauer in der REST-Tabelle beschrieben. Wenn Daten unter den Systemkomponenten ausgetauscht werden, dann immer in Form einer JSON-Objekts.

## Spendenorganisationen

- Save Our Seeds (SOS)
- Bund
- Albert Schweizer Stiftung für unsere Mitwelt

Save Our Seeds (SOS)  
[saveourseeds.org](https://saveourseeds.org)

SOS engagiert sich seit 2002 für die Reinhaltung des Saatguts von jeglichen gentechnischen Verunreinigungen („Nulltoleranz“) in Europa. Die Agrar-Gentechnik gilt als unkontrollierbare Risikotechnologie, darum will SOS ihren Einsatz verhindern und gleichzeitig die Vielfalt unseres Saatguts und die gentechnikfreie Landwirtschaft erhalten. Die sinnvolle Initiative wurde von der Zukunftsstiftung Landwirtschaft gegründet und wird derzeit von rund 300 Organisationen und 300.000 Menschen unterstützt.

Bund  
[bund.net](https://bund.net)

Der BUND setzt sich bereits seit über 40 Jahren für den Naturschutz ein und ist heute einer der wichtigsten deutschen Umweltverbände. Eines seiner zentralen Anliegen ist eine ökologische Landwirtschaft – inklusive einer Verbesserung des Tierschutzes: „Der BUND fordert mehr Naturschutz in der Landwirtschaft und das Ende tierquälerischer Haltungsformen. Der Missbrauch von Antibiotika in der Massentierhaltung muss gestoppt werden – damit sich Verbraucher auf einwandfreie Lebensmittel verlassen können.“ Mit einer Geldpende unterstützt du also den Kampf gegen Massentierhaltung.

Albert Schweitzer Stiftung für unsere Mitwelt

Die „Albert Schweitzer Stiftung für unsere Mitwelt“ ist eine Tierschutzorganisation mit einer Vision: „die Abschaffung der Massentierhaltung und eine weite Verbreitung der veganen Lebensweise“. Um diesem Ziel näher zu kommen, organisiert die Stiftung Kampagnen und Petitionen und informiert Konsumenten über Alternativen – etwa mit dem Projekt Vegan Taste Week. Dabei bekommen Neugierige kostenlos Tipps, Infos und Rezepte rund um veganes Leben.

[albert-schweitzer-stiftung.de](https://albert-schweitzer-stiftung.de)