

PFLICHTENHEFT

FutureMarket – Pflichtenheft

Digitaler Einkaufswagen

Webanwendung Digitaler Einkaufswagen

001

Hinweis: Bitte Beschriften Sie Tabellen und Abbildungen um diese anschließend in einem Verzeichnis aufzulisten.

Änderungshistorie

Version	Datum	Änderung	Name
0.1	13.10.2021	Erstellung	Max Muster
0.2	23.11.2021	Anpassung	Mara Müller
0.3	07.01.2022	Technische Inhalte	Max Muster
0.4	18.01.2022	Technische Inhalte	Max Muster

Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

PFLICHTENHEFT

Herausgeber	Max Muster		
Dateiname	Musterlösung_Pflichtenheft_FutureMarket_0.4		
Dokumentennummer	001		
Dokumentenbezeichnung	FutureMarket - Pflichtenheft		
Version	0.4		
Stand	18.01.2022		
Status	Aktiv		
Autor	Max Muster		
Inhaltlich geprüft von	Mara Müller		
Freigegeben von	Mara Mustermann		
Ansprechpartner	Max Muster	+491256893	max@muster.de
Kurzinfo	FutureMarket Pflichtenheft für Webanwendung Digitaler Einkaufswagen		

PFLICHTENHEFT

Inhalt

1	Ausgangssituation.....	6
2	Aufgabenstellung	6
2.1	Wunschkriterien	7
3	Implementierungsentwurf	7
3.1	Gesamtsystem	7
3.2	Implementierung von Komponente <Zentrales Backend>	8
3.2.1	Klassendiagramm Komponente <Zentrales Backend>	8
3.3	Implementierung von Komponente <App>	11
3.3.1	Klassendiagramm Komponente <App>	11
3.4	Implementierung von Komponente <Einkaufswagen>	15
3.4.1	Klassendiagramm Komponente <Einkaufswagen>	15
3.5	Anbindung von Komponente <Kassenbereich>	17
3.6	Anbindung von Komponente <Ausgangsschranke>	17
4	Funktionale (Detail-) Anforderungen	19
4.1	Anforderungspaket Geschäftsfälle	19
4.2	Detailanforderung: Geschäftsfall 2	20
4.3	Detailanforderung: Geschäftsfall 3	22
4.4	Detailanforderung: Geschäftsfall 7 + 8	23
5	Messages	24
6	Technische Architektur	25
6.1	Systemarchitektur	25
6.2	Verwendete Technologien	25
7	Nichtfunktionale Anforderungen	25
7.1	Zuverlässigkeit	26
7.2	Benutzerfreundlichkeit	26
8	Beteiligte	27
8.1	Kunde/Auftraggeber/Interne Beteiligte	27
8.2	Anwender	27
9	Umfang der Anforderung.....	27
10	Offene Punkte.....	28
11	Testanforderungen	28
12	Abkürzungsverzeichnis.....	29

PFLICHTENHEFT

Abbildungsverzeichnis

Abbildung 1: Komponentendiagramm des Gesamtsystems	7
Abbildung 2: Klassendiagramm der Komponente <Zentrales Backend>.....	8
Abbildung 3: Klassendiagramm Komponente <APP>	11
Abbildung 4: Klassendiagramm der Komponente <Einkaufswagen>	16
Abbildung 5: Flussdiagramm – Registrieren	20
Abbildung 6: Flussdiagramm – Verbinden von Wagen und Kundenkonto	22
Abbildung 7: Flussdiagramm - Bezahlen und Schranke.....	23
Abbildung 8: Systemarchitektur	25

PFLICHTENHEFT

Tabellenverzeichnis

Table 9: Anforderungspaket Geschäftsfälle	19
Table 10: Definierte Messages	24
Table 12: Kunde/Auftraggeber/Interne Beteiligte	27
Table 13: Anwender	27
Table 14: Umfang der Anforderung.....	27
Table 15: Offene Punkte.....	28
Table 16: Abkürzungsverzeichnis	29

PFLICHTENHEFT

1 Ausgangssituation

Das Unternehmen FutureMarket verfügt zum heutigen Zeitpunkt über ein analoges Kassensystem. Das analoge Kassensystem basiert darauf, dass der Kunde sich zunächst einen Einkaufswagen holt. Während des Einkaufens legt er seine Waren dort hinein. Sobald sich alle Produkte in dem Einkaufswagen befinden, geht er zur Kasse. Dort legt er alle Produkte aus dem Einkaufswagen auf das Kassenband. Die Produkte werden von dem Kassenspersonal gescannt. Am Ende bezahlt der Kunde bar oder mit Karte. An manchen Standorten sind Selbstscann-Kassen vorhanden.

2 Aufgabenstellung

Um das analoge Kassensystem abzuschaffen und das Einkaufen zu digitalisieren, fordert das Unternehmen FutureMarket einen digitalen Einkaufswagen, eine App für den Kunden und einen digitalen Kassenbereich inklusive einer Ausgangsschranke.

Kunden sollen die App auf ihr Smartphone aus den bekannten AppStores herunterladen können. Innerhalb der App soll es möglich sein ein Konto zu erstellen und ein Zahlungsmittel zu hinterlegen. Als Zahlungsmittel sind Kreditkarte, Girokonto und PayPal erlaubt. Um den Einkaufsprozess zu starten, verbindet der Kunde sein Kundenkonto mit dem digitalen Einkaufswagen, indem er die App öffnet und sein Handy kurz auf den Scanner am digitalen Einkaufswagen legt. Der Einkaufswagen verfügt über einen NFC Chip, welcher die Verbindung von Einkaufswagen und Kundenkonto herstellt. Der Einkaufswagen muss Waren sowie deren Gewicht und Menge automatisch erkennen. Des Weiteren soll die Möglichkeit existieren bargeldlos zu bezahlen.

Der Einkaufswagen verfügt über Kameras, die beim Einlegen von Produkten diese erkennen. Die Produkte (inklusive Gewicht/ Menge) werden innerhalb der App des Kunden registriert. Dafür beinhaltet die App einen digitalen Einkaufszettel. Falls Artikel aus dem Wagen herausgenommen werden, erkennt dies der Wagen und die Artikel werden von dem digitalen Einkaufszettel entfernt. Außerdem verfügt der Einkaufswagen über eine Waage, welche das Wiegen von losen Artikeln bspw. Obst und Gemüse ermöglicht.

Des Weiteren verfügt der Einkaufswagen über einen Touchscreen. Dieser ermöglicht das manuelle Eingeben eines EAN Codes. Dies wird benötigt, falls der Einkaufswagen einen Artikel nicht erkennt. Der Kunde kann den Artikel dann manuell hinzufügen.

Zusätzlich soll ein Kassenbereich bereitgestellt werden. Dieser Bereich ist markiert und wird betreten, sobald der Kunde alle Artikel in den Einkaufswagen gelegt hat. Hinter dem Kassenbereich soll sich die Ausgangsschranke befinden.

Die Endsumme aller im Einkaufswagen vorhanden Artikel wird berechnet und abgerechnet (vom hinterlegten Zahlungsmittel) sobald der Kunde sich mit dem Wagen in dem Kassenbereich befindet. Bei erfolgreicher Bezahlung öffnet sich die Ausgangsschranke. Der Kunde kann den Markt verlassen und der Einkaufswagen wird automatisch zurückgesetzt. Bei nicht erfolgreicher Zahlung soll eine rote Lampe aufleuchten und die Schranke soll geschlossen bleiben.

Die Mitarbeiter im Laden sollen nun benachrichtigt werden. Zusätzlich muss der Kunde den Einkauf nun bei einem Mitarbeiter abschließen.

PFLICHTENHEFT

2.1 Wunschkriterien

Berücksichtigung gängiger Normen und Richtlinien der Softwareergonomie.

3 Implementierungsentwurf

3.1 Gesamtsystem

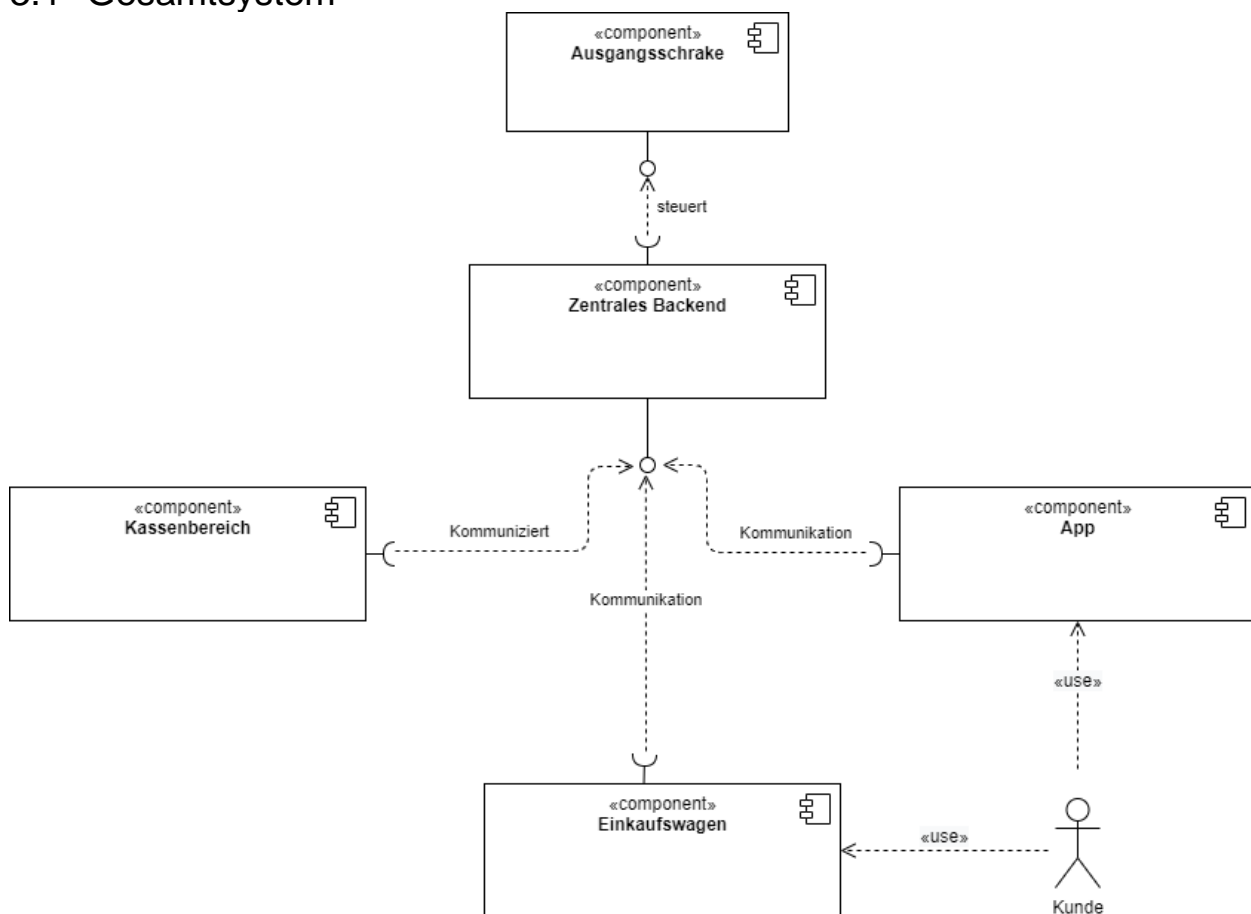


Abbildung 1: Komponentendiagramm des Gesamtsystems

Das Gesamtsystem besteht aus den folgenden Komponenten:

- Zentrales Backend
- App (Smartphone Software)
- Einkaufswagen (Software)
- Kassenbereich. Das zentrale Backend stellt eine REST konforme Schnittstelle zur Verfügung über die der Kassenbereich die nötigen Daten übermitteln kann.
- Ausgangsschranke. Es wird davon ausgegangen, dass die Ausgangsschranke eine REST konforme Schnittstelle zur Verfügung stellt, damit das zentrale Backend die Schranke öffnen und schließen kann.

Hinweis: Die Software für den Kassenbereich sowie die Ausgangsschranke ist aktuell **nicht Gegenstand des Pflichtenheftes!** Auf Wunsch kann diese jedoch zusätzlich beauftragt werden.

PFLICHTENHEFT

3.2 Implementierung von Komponente <Zentrales Backend>

Die Komponente <Zentrales Backend> stellt die notwendigen Funktionen und Schnittstellen bereit, um die Komponenten <App>, <Einkaufswagen>, <Kassenbereich> und <Ausgangsschranke> korrekt miteinander zu verbinden.

3.2.1 Klassendiagramm Komponente <Zentrales Backend>

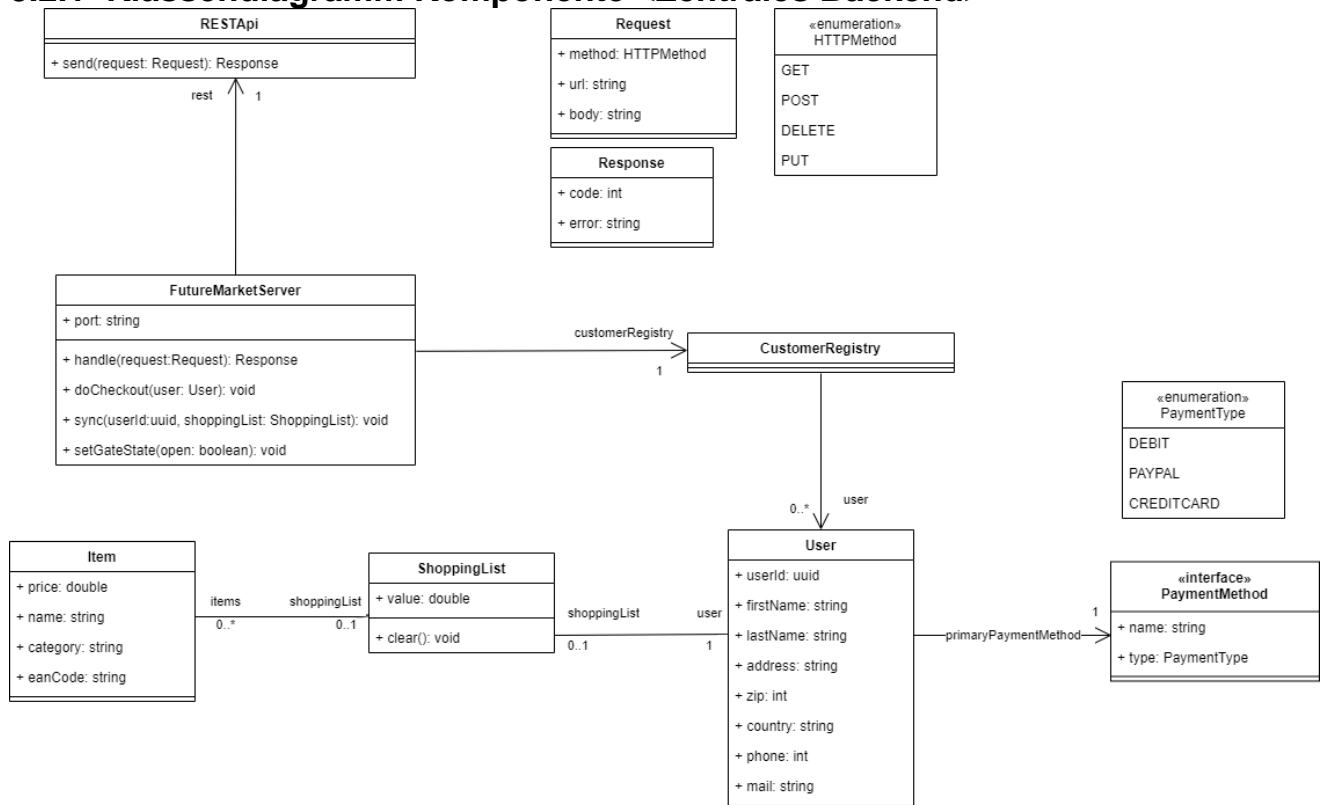


Abbildung 2: Klassendiagramm der Komponente <Zentrales Backend>

3.2.1.1 Beschreibung der Methoden der Klasse RESTApi

Methodenname	Parameter	Rückgabewert	Beschreibung
send	request:Request	Response	Sendet den als Parameter übergebenen Request über das Netzwerk.

3.2.1.2 Beschreibung der Attribute der Klasse Request

Attributname	Typ	Beschreibung
method	HTTPMethod	HTTP Methode.
url	String	URL des Requests.
body	String	Der Request body.

3.2.1.3 Beschreibung der Attribute der Klasse Response

Attributname	Typ	Beschreibung
code	int	HTTP Statuscode.

PFLICHTENHEFT

error	String	Error, falls verfügbar.
-------	--------	-------------------------

3.2.1.4 Beschreibung der Attribute der Enumeration HTTPMethod

Attributname	Beschreibung
GET	HTTP Methode GET.
POST	HTTP Methode POST.
DELETE	HTTP Methode DELETE.
PUT	HTTP Methode PUT.

3.2.1.5 Beschreibung der Attribute und Methoden der Klasse FutureMarketServer

Attributname	Typ	Beschreibung
port	int	Der Port auf dem der HTTP Server laufen soll.
rest	RESTApi	Referenz auf ein RESTApi Objekt für die Netzwerkkommunikation.
customerRegistry	CustomerRegistry	Referenz auf ein CustomerRegistry Objekt um die aktuell im Markt einkaufenden Kunden zu verwalten.

Methodenname	Parameter	Rückgabewert	Beschreibung
handle	request:Request	Response	Verarbeitet den als Parameter übergebenen Request der über das Netzwerk empfangen wurde. Liefert eine passende Response zurück.
doCheckout	user:User	void	Initiiert den Abrechnungsprozess für den User mit der übergebenen ID.
sync	userId:uuid shoppingList: ShoppingList	void	Synchronisiert den übergebenen Warenkorb für den User mit der übergebenen ID.
setGateState	open: boolean	void	Sendet einen Request an die Ausgangsschranke um den aktuellen Zustand der Schranke zu ändern.

3.2.1.6 Beschreibung der Attribute und Methoden der Klasse User

Attributname	Typ	Beschreibung
userId	uuid	Eindeutige ID des Benutzers.
firstName	string	Der Vorname des Benutzers.
lastName	string	Der Nachname des Benutzers.

PFLICHTENHEFT

address	string	Die Adresse des Benutzers.
zip	string	Die Postleitzahl des Benutzers.
country	string	Das Herkunftsland des Benutzers.
phone	int	Die Telefonnummer des Benutzers.
mail	string	Die E-Mail Adresse des Benutzers.
primaryPaymentmethod	PaymentMethod	Die primäre Zahlungsmethode des Benutzers.
paymentMethods	List<PaymentMethod>	Liste aller Zahlungsmethoden des Benutzers.
shoppingList	ShoppingList	Der digitale Einkaufszettel des Benutzers.

3.2.1.7 Beschreibung der Attribute und Methoden der Klasse ShoppingList

Attributname	Typ	Beschreibung
value	double	Der aktuelle Warenwert des digitalen Einkaufszettels.
items	List<Item>	Liste aller Gegenstände im Einkaufswagen.
user	User	Referenz auf den Benutzer der diesem digitalen Einkaufszettel zugeordnet ist.

Methodenname	Parameter	Rückgabewert	Beschreibung
clear	-	void	Entfernt alle Artikel von dem digitalen Einkaufszettel und setzt den Warenwert auf 0.

3.2.1.8 Beschreibung der Attribute der Klasse Item

Attributname	Typ	Beschreibung
price	double	Der Preis des Artikels.
name	string	Der Artikelname.
category	string	Die Produktkategorie des Artikels.
eanCode	string	Der EAN Code des Artikels.
shoppingList	ShoppingList	Der digitale Einkaufszettel dem dieser Artikel zugeordnet ist.

3.2.1.9 Beschreibung der Attribute des Interface PaymentMethod

Attributname	Typ	Beschreibung
name	string	Der Name der Zahlungsmethode.
type	PaymentType	Der Typ der Zahlungsmethode.

3.2.1.10 Beschreibung der Attribute der Enumeration PaymentType

Attributname	Beschreibung
DEBIT	Typ der Zahlungsmethode Bankeinzug.

PFLICHTENHEFT

PAYPAL	Typ der Zahlungsmethode Paypal.
CREDIT	Typ der Zahlungsmethode Kreditkarte.

3.3 Implementierung von Komponente <App>

Die Komponente <App> umfasst die Funktionalität des digitalen Einkaufszettels sowie die Verwaltung des Kundenkontos. Die App kann über eine REST Schnittstelle mit dem zentralen Backend kommunizieren.

Die App kann auf aktuellen Geräten (>2021) mit den Betriebssystemen iOS und Android verwendet werden.

3.3.1 Klassendiagramm Komponente <App>

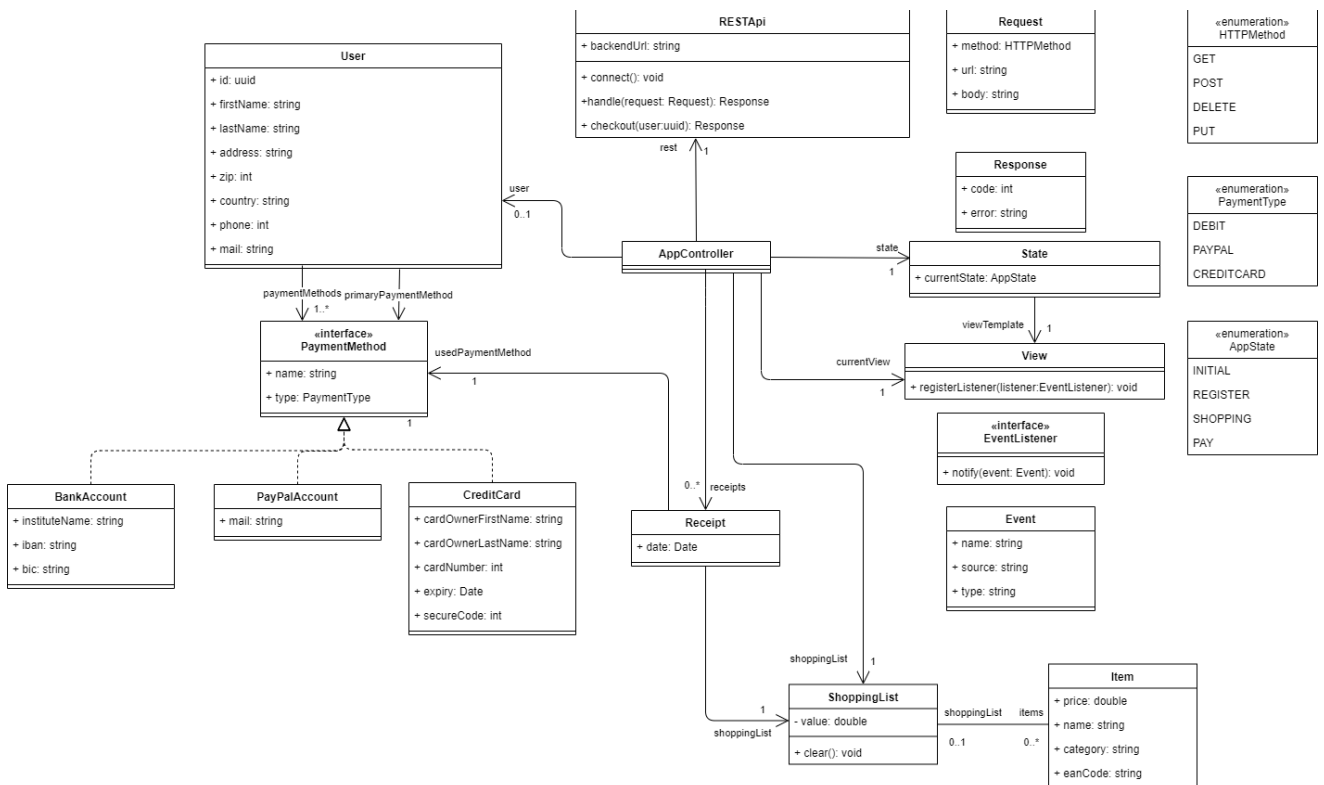


Abbildung 3: Klassendiagramm Komponente <APP>

3.3.1.1 Beschreibung der Attribute und Methoden der Klasse User

Attributname	Typ	Beschreibung
userId	uuid	Eindeutige ID des Benutzers.
firstName	string	Der Vorname des Benutzers.
lastName	string	Der Nachname des Benutzers.
address	string	Die Adresse des Benutzers.
zip	string	Die Postleitzahl des Benutzers.
country	string	Das Herkunftsland des Benutzers.

PFLICHTENHEFT

phone	int	Die Telefonnummer des Benutzers.
mail	string	Die E-Mail Adresse des Benutzers.

3.3.1.2 Beschreibung der Attribute des Interface PaymentMethod

Attributname	Typ	Beschreibung
name	string	Der Name der Zahlungsmethode.
type	PaymentType	Der Typ der Zahlungsmethode.

3.3.1.3 Beschreibung der Attribute der Enumeration PaymentType

Attributname	Beschreibung
DEBIT	Typ der Zahlungsmethode Bankeinzug.
PAYPAL	Typ der Zahlungsmethode Paypal.

3.3.1.4 Beschreibung der Attribute und Methoden der Klasse BankAccount

Attributname	Typ	Beschreibung
instituteName	string	Der Name des Kreditinstituts.
iban	string	Die IBAN der Bankverbindung.
bic	string	Die BIC der Bankverbindung.

3.3.1.5 Beschreibung der Attribute und Methoden der Klasse PayPalAccount

Attributname	Typ	Beschreibung
mail	string	Die E-Mail Adresse die zum Anmelden am PayPal Konto benutzt wird.

3.3.1.6 Beschreibung der Attribute und Methoden der Klasse CreditCard

Attributname	Typ	Beschreibung
cardOwnerFirstName	string	Der Vorname des Karteninhabers.
cardOwnerLastName	string	Der Nachname des Karteninhabers.
cardNumber	int	Die Kreditkartennummer.
expiry	Date	Das Ablaufdatum der Kreditkarte.
secureCode	int	Der Sicherheitscode der Kreditkarte.

3.3.1.7 Beschreibung der Attribute und Methoden der Klasse RESTApi

PFLICHTENHEFT

Attributname	Typ	Beschreibung
backendUrl	string	Die URL zum zentralen FutureMarket Backend.

Methodenname	Parameter	Rückgabewert	Beschreibung
connect	-	void	Methode um eine Anmeldung am Einkaufswagen vorzunehmen.
handle	request:Request	Response	Verarbeitet den als Parameter übergebenen Request der über das Netzwerk empfangen wurde. Liefert eine passende Response zurück.
checkout	user:uuid	Response	Initiiert den Bezahlprozess in der Kassenzone für den Nutzer mit der übergebenen userId.

3.3.1.8 Beschreibung der Attribute der Klasse Request

Attributname	Typ	Beschreibung
method	HTTPMethod	HTTP Methode.
url	String	URL des Requests.
body	String	Der Request body.

3.3.1.9 Beschreibung der Attribute der Klasse Response

Attributname	Typ	Beschreibung
code	int	HTTP Statuscode.
error	String	Error, falls verfügbar.

3.3.1.10 Beschreibung der Attribute der Enumeration HTTPMethod

Attributname	Beschreibung
GET	HTTP Methode GET.
POST	HTTP Methode POST.
DELETE	HTTP Methode DELETE.
PUT	HTTP Methode PUT.

3.3.1.11 Beschreibung der Attribute der Klasse AppController

Attributname	Typ	Beschreibung
user	User	Der aktuell eingeloggte Benutzer.
rest	RESTApi	Referenz auf ein RESTApi Objekt für den Netzwerkzugriff.

PFLICHTENHEFT

state	State	Referenz auf den aktuellen Zustand der Applikation.
currentView	View	Referenz auf die aktuelle Benutzeroberfläche der Applikation.
receipts	List<Receipt>	Referenz auf alle digitalen Kassenbons.
shoppingList	ShoppingList	Referenz auf den aktuellen digitalen Einkaufszettel.

3.3.1.12 Beschreibung der Attribute der Klasse State

Attributname	Typ	Beschreibung
currentState	AppState	Der Typ des Zustands.
viewTemplate	View	Referenz auf eine View die erzeugt wird, sobald der Zustand erreicht wird.

3.3.1.13 Beschreibung der Attribute der Enumeration AppState

Attributname	Beschreibung
INITIAL	Der Initialzustand der Applikation. In diesem kann der Benutzer sich entweder anmelden oder einen neuen Account registrieren.
REGISTER	Zustand während der Restrierung.
SHOPPING	Zustand während des Einkaufs.
PAY	Zustand während des Bezahlvorgangs.

3.3.1.14 Beschreibung der Methoden der Klasse View

Methodenname	Parameter	Rückgabewert	Beschreibung
registerListener	Listener: EventListener	void	Registriert einen EventListener an der View um über Nutzerevents informiert zu werden.

3.3.1.15 Beschreibung der Methoden des Interface EventListener

Methodenname	Parameter	Rückgabewert	Beschreibung
notify	event:Event	void	Informiert den EventListener, dass das als Parameter übergebene Event aufgetreten ist.

3.3.1.16 Beschreibung der Attribute der Klasse Event

Attributname	Typ	Beschreibung
name	string	Der Name des Events.
source	string	Die Quelle des Events.
type	string	Der Typ des Events.

3.3.1.17 Beschreibung der Attribute der Klasse Receipt

Attributname	Typ	Beschreibung
--------------	-----	--------------

PFLICHTENHEFT

date	string	Der Name des Events.
shoppingList	ShoppingList	Referenz auf den digitalen Einkaufszettel des Kassensbons.

3.3.1.18 Beschreibung der Attribute und Methoden der Klasse ShoppingList

Attributname	Typ	Beschreibung
value	double	Der aktuelle Warenwert des digitalen Einkaufszettels.
items	List<Item>	Liste aller Gegenstände im Einkaufswagen.

Methodenname	Parameter	Rückgabewert	Beschreibung
clear	-	void	Entfernt alle Artikel von dem digitalen Einkaufszettel und setzt den Warenwert auf 0.

3.3.1.19 Beschreibung der Attribute der Klasse Item

Attributname	Typ	Beschreibung
price	double	Der Preis des Artikels.
name	string	Der Artikelname.
category	string	Die Produktkategorie des Artikels.
eanCode	string	Der EAN Code des Artikels.
shoppingList	ShoppingList	Der digitale Einkaufszettel dem dieser Artikel zugeordnet ist.

3.4 Implementierung von Komponente <Einkaufswagen>

Die Komponente <Einkaufswagen> umfasst die Software für den Betrieb des Touchscreens sowie die Erfassung der Waren im Einkaufswagen. Zudem wird sichergestellt, dass der aktuelle Zustand jederzeit an die gerade gekoppelte App gesendet wird.

3.4.1 Klassendiagramm Komponente <Einkaufswagen>

PFLICHTENHEFT

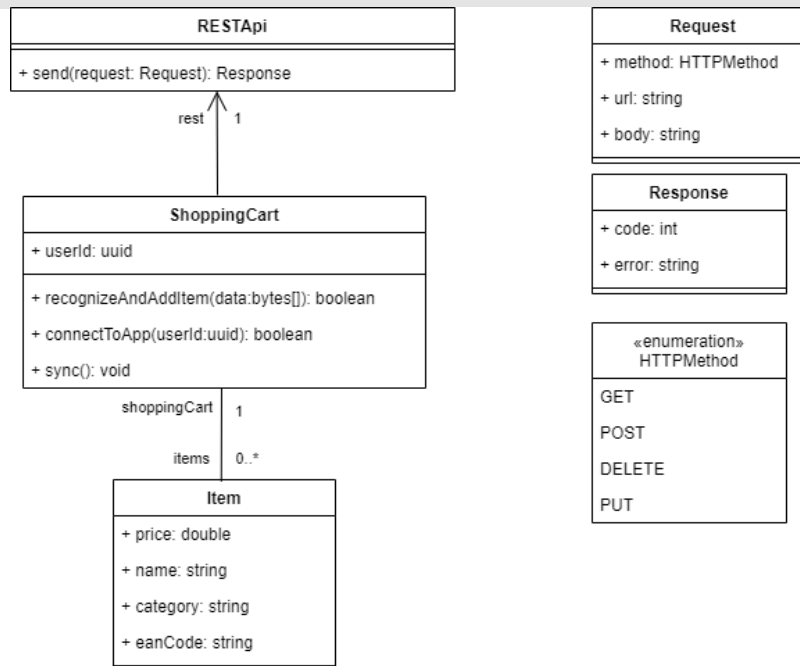


Abbildung 4: Klassendiagramm der Komponente <Einkaufswagen>

3.4.1.1 Beschreibung der Methoden der Klasse RESTApi

Methodenname	Parameter	Rückgabewert	Beschreibung
send	request: Request	Response	Sendet den als Parameter übergebenen Request über das Netzwerk.

3.4.1.2 Beschreibung der Attribute der Klasse Request

Attributname	Typ	Beschreibung
method	HTTPMethod	HTTP Methode.
url	String	URL des Requests.
body	String	Der Request body.

3.4.1.3 Beschreibung der Attribute der Klasse Response

Attributname	Typ	Beschreibung
code	int	HTTP Statuscode.
error	String	Error, falls verfügbar.

3.4.1.4 Beschreibung der Attribute der Enumeration HTTPMethod

Attributname	Beschreibung
GET	HTTP Methode GET.
POST	HTTP Methode POST.
DELETE	HTTP Methode DELETE.

PFLICHTENHEFT

3.4.1.5 Beschreibung der Attribute der Klasse Item

Attributname	Typ	Beschreibung
price	double	Der Preis des Artikels.
name	string	Der Artikelname.
category	string	Die Produktkategorie des Artikels.
eanCode	string	Der EAN Code des Artikels.
shoppingCart	ShoppingCart	Referenz auf den Einkaufswagen in dem sich der Artikel befindet.

3.4.1.6 Beschreibung der Attribute und Methoden der Klasse ShoppingCart

Attributname	Typ	Beschreibung
userId	uuid	Die eindeutige ID des Benutzers.

Methodenname	Parameter	Rückgabewert	Beschreibung
recognizeAndAddItem	data:byte[]	boolean	Erkennt den im Binärformat über den Parameter data übergebenen Artikel und fügt ihn dem Einkaufswagen hinzu. Gibt true zurück, falls der Vorgang erfolgreich war, false andernfalls.
connectToApp	userId:uuid	boolean	Verbindet den Einkaufswagen mit der App eines Benutzers. Gibt true zurück, falls der Vorgang erfolgreich war, false andernfalls. Registriert die Session gleichzeitig beim zentralen FutureMarket Backend.
sync	-	void	Sendet den aktuellen Einkaufswagen des Nutzers an das zentrale FutureMarket backend. Diese Methode wird nach jeder Änderung des Einkaufswagens aufgerufen.

3.5 Anbindung von Komponente <Kassenbereich>

Für die Komponente <Kassenbereich> wird seitens des zentralen Backends eine REST basierte Schnittstelle zur Verfügung gestellt, damit der Kassenbereich eine Bezahlanforderung für einen bestimmten Nutzer initiieren kann.

Siehe Komponente <Zentrales Backend> für Details hinsichtlich der Anbindung des Kassenbereichs.

3.6 Anbindung von Komponente <Ausgangsschranke>

Die Komponente <Ausgangsschranke> wird über eine REST konforme Schnittstelle

PFLICHTENHEFT

angesprochen, um das Öffnen und Schließen seitens des zentralen Backends zu ermöglichen.

Siehe Komponente <Zentrales Backend> für Details hinsichtlich der Anbindung des Kassenbereichs.

PFLICHTENHEFT

4 Funktionale (Detail-) Anforderungen

4.1 Anforderungspaket Geschäftsfälle

Geschäftsfälle Sicht Kunde	
1	Herunterladen der App
2	Registrierung in der App (inkl. Zahlungsmittel)
3	Verbinden von Wagen und Kundenkonto
4	Einkaufen (Ware mit Barcode)
5	Einkaufen (Lose Ware)
6	Artikel entfernen
7	Zahlen
8	Schranke öffnet

Table 1: Anforderungspaket Geschäftsfälle

PFLICHTENHEFT

4.2 Detailanforderung: Geschäftsfall 2

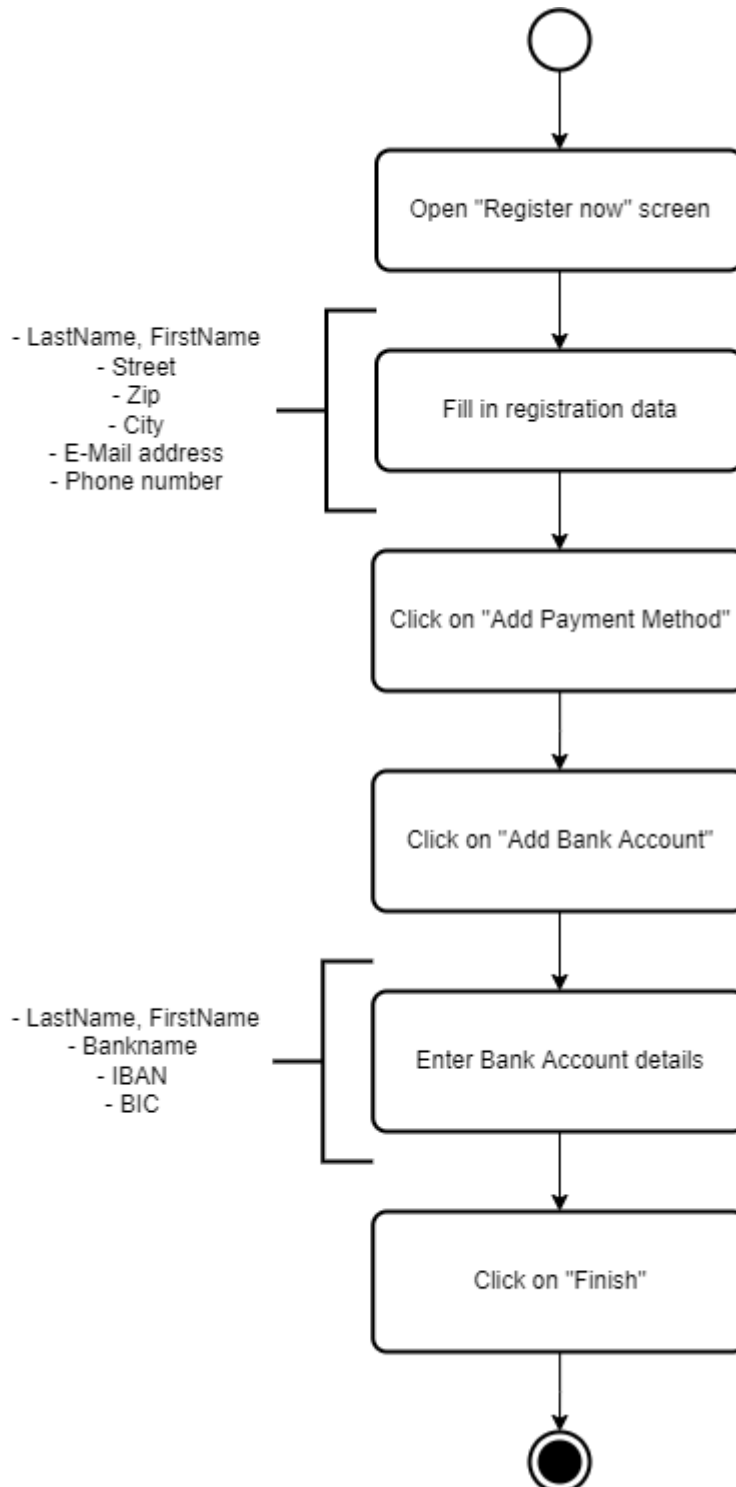


Abbildung 5: Flussdiagramm – Registrieren

PFLICHTENHEFT

Beteiligte Klassen	Methode	Beschreibung	Notwendige Attribute	Kommunikationspartner		
AppController	setState()	Ruft den Registrierbildschirm auf	state:AppState	n/a		
RESTApi	handleRequest()	Schickt die Registrierung an das FutureMarket Backend	request:Request	<table border="1"> <tr> <td>Name der Komponente</td> </tr> <tr> <td>Zentrales Backend</td> </tr> </table>	Name der Komponente	Zentrales Backend
Name der Komponente						
Zentrales Backend						

PFLICHTENHEFT

4.3 Detailanforderung: Geschäftsfall 3

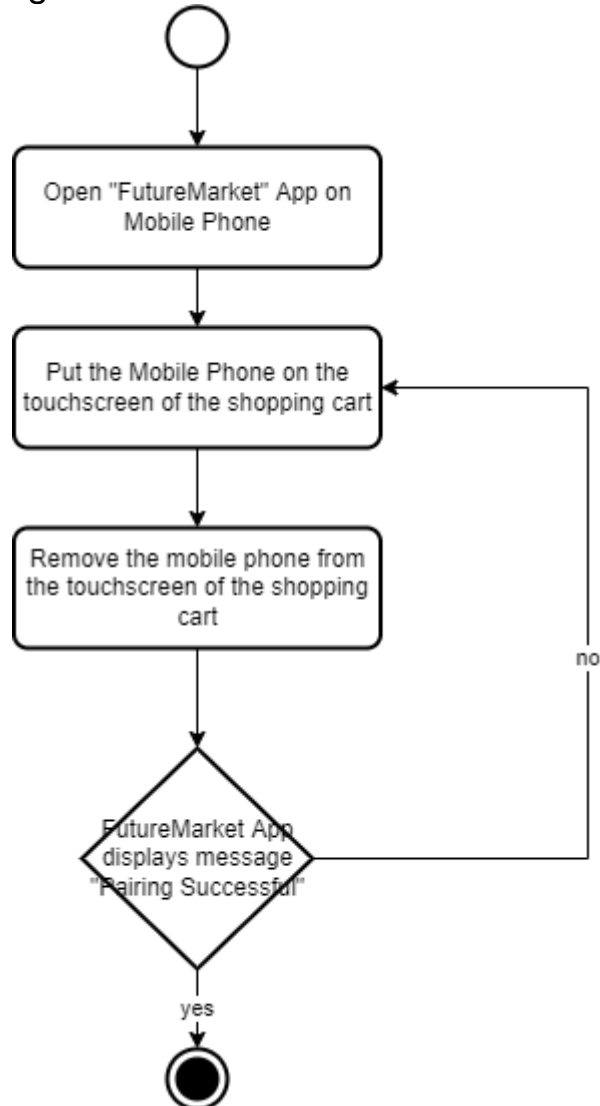


Abbildung 6: Flussdiagramm – Verbinden von Wagen und Kundenkonto

Beteiligte Klassen	Methode	Beschreibung	Notwendige Attribute	Kommunikationspartner		
AppController	setState()	Ruft den Pairing Bildschirm auf	state:AppState	n/a		
RESTApi	handleRequest()	Schickt eine Pairing Anfrage an den Einkaufswagen	request:Request	<table border="1"> <tr> <td>Name der Komponente</td> </tr> <tr> <td>Einkaufswagen</td> </tr> </table>	Name der Komponente	Einkaufswagen
Name der Komponente						
Einkaufswagen						

PFLICHTENHEFT

4.4 Detailanforderung: Geschäftsfall 7 + 8

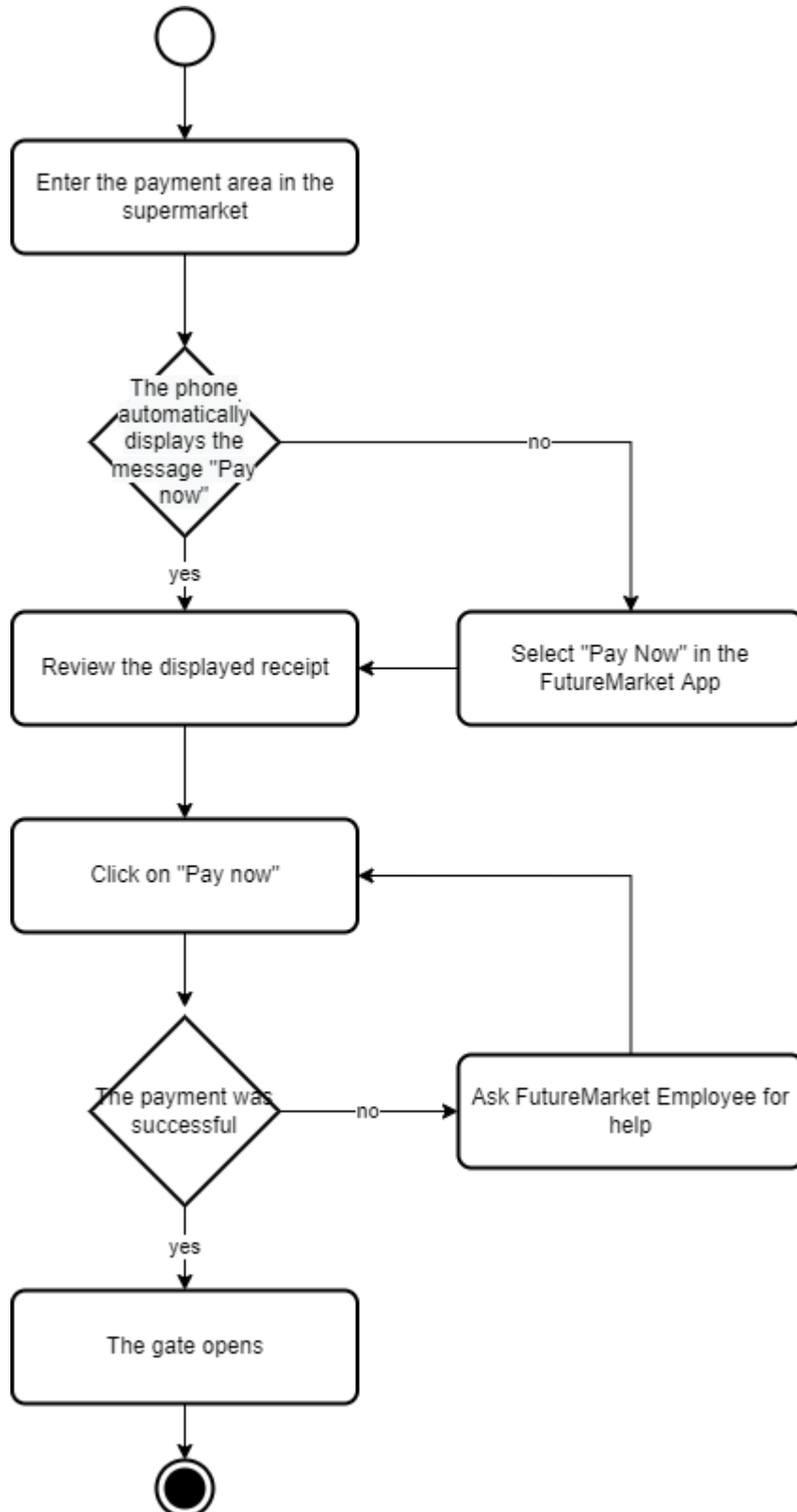


Abbildung 7: Flussdiagramm - Bezahlen und Schranke

PFLICHTENHEFT

Beteiligte Klassen	Methode	Beschreibung	Notwendige Attribute	Kommunikationspartner		
AppController	setState()	Ruft den Bezahlbildschirm auf	state:AppState	n/a		
RESTApi	handleRequest()	Schickt eine Bezahlfrage Anfrage an das zentrale Backend	request:Request	<table border="1"> <tr> <td>Name der Komponente</td> </tr> <tr> <td>Zentrales Backend</td> </tr> </table>	Name der Komponente	Zentrales Backend
Name der Komponente						
Zentrales Backend						

5 Messages

Geschäftsfall	Fehler	Feld	Art der Meldung	Message
Registrieren in der App	Nein		I	Registrierung erfolgreich. Viel Spaß beim Einkaufen.
Verbinden von Wagen und Kundenkonto	Ja		E	Entschuldigung, da hat etwas nicht geklappt. Bitte versuche es noch einmal.
Zahlen	Nein		I	Deine Zahlung per Bankkonto war erfolgreich
Schranke öffnet	Nein		I	Die Schranke ist nun geöffnet, hab einen schönen Tag.

Table 2: Definierte Messages

Zeichen	Bedeutung
E	Fehler
W	Warnung
I	Information

PFLICHTENHEFT

6 Technische Architektur

6.1 Systemarchitektur

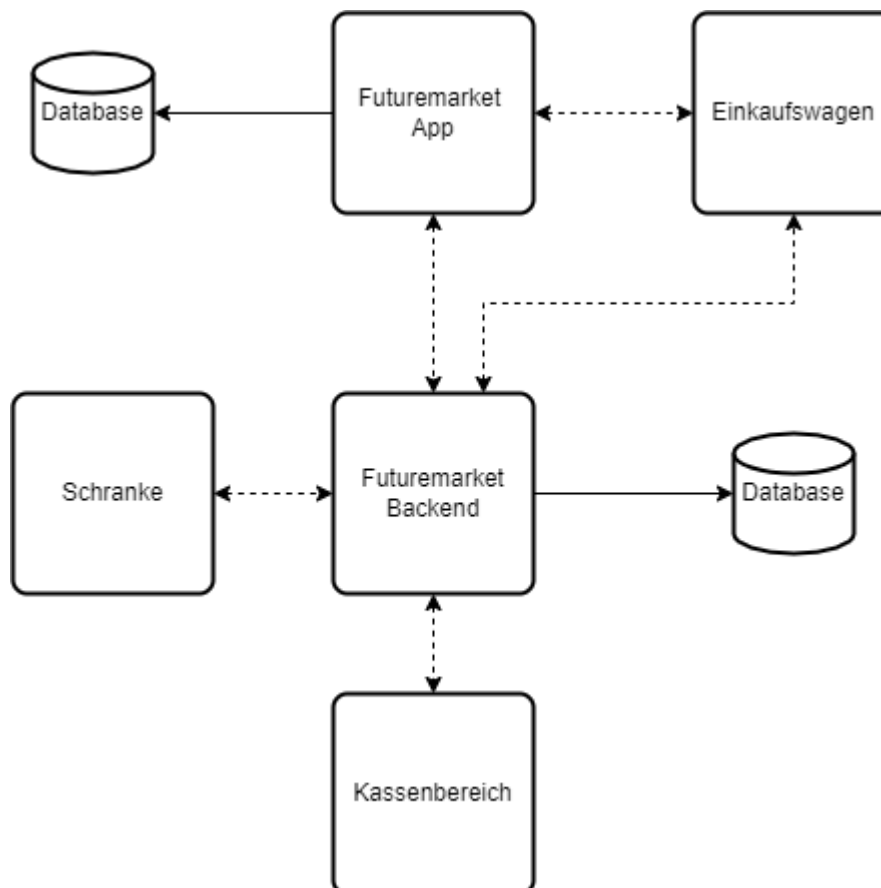


Abbildung 8: Systemarchitektur

6.2 Verwendete Technologien

- FutureMarket App
 - o Kotlin Mobile SDK für Cross-Plattform Entwicklung
- FutureMarket Backend
 - o Node.js
 - o Typescript
 - o SAP CAP
 - o Maria DB
 - o React für die Administrationsoberfläche des Backends

7 Nichtfunktionale Anforderungen

Das Endprodukt soll den EN ISO 9241-11 Standard erfüllen. Ziel ist es, eine effektive, effiziente und zufriedenstellende Anwendung zu erstellen. Hierbei soll zusätzlich ein besonderes Augenmerk auf die Zuverlässigkeit, Wartbarkeit sowie die Skalierbarkeit gelegt werden.

PFLICHTENHEFT

7.1 Zuverlässigkeit

Das System muss jederzeit (bei jedweder Belastungsgröße) zuverlässig reagieren. Es darf nicht zu unkontrollierten Systemabstürzen oder Datenverlust kommen. Programme und Daten müssen gegen zufällige und unabsichtliche Veränderungen geschützt werden.

7.2 Benutzerfreundlichkeit

Der Anwender muss zeitnah (Antwortzeit < 0,5s) auf Fehler und falsche Eingaben hingewiesen werden. Er muss durch eine Hilfefunktion bei der Anwendung unterstützt werden. Die graphischen Oberflächen müssen übersichtlich, einheitlich strukturiert und robust sein und die geforderte Funktionalität anbieten. Sie müssen intuitiv bedienbar sein, das heißt der Anwender muss ohne Schulung, also nur mit der angebotenen Hilfefunktion, fähig sein mit dem System umzugehen. Bei der Erstellung der grafischen Oberflächen sind des Weiteren die gängigen Normen und Richtlinien der Softwareergonomie zu berücksichtigen. Die Erstellung einer barrierefreien Benutzeroberfläche ist nicht zwingend gefordert.

PFLICHTENHEFT

8 Beteiligte

8.1 Kunde/Auftraggeber/Interne Beteiligte

Ansprechpartner	Name	OrgE	Telefon
Alle Themen	Max Muster	OctaVIA AG	Siehe Adressbuch
FutureMarket Themen	Jens Heine	FutureMarket	Siehe Adressbuch

Table 3: Kunde/Auftraggeber/Interne Beteiligte

8.2 Anwender

Nutzer	Geschätzte Anzahl an Personen	Nutzungsart
Kunden	Ca. 35 gleichzeitig (pro Markt)	Einkaufen
Kunden	6 gleichzeitig	Bezahlen

Table 4: Anwender

9 Umfang der Anforderung

Komponente	Zeit	Entwickler
Zentrales Backend	3 Monate	3
App	2 Monate	3
Einkaufswagen	Diese Komponenten werden zugeliefert.	
Ausgangsschranke		
Kassenbereich		

Table 5: Umfang der Anforderung

PFLICHTENHEFT

10 Offene Punkte

Kapitel / Seite	OPP/Anmerkung	Beschluss	Art	Verantwortlich	Erledigungsdatum

Table 6: Offene Punkte

11 Testanforderungen

Das System wird nach den Kriterien der ISO/IEC 25000 Norm geprüft, welche als Leitfaden für die Bewertung von Softwarequalität gilt. Somit werden die Qualität, Funktionalität und Sicherheit der Software getestet und sichergestellt.

Mithilfe funktionaler Tests wird das System auf Korrektheit und Vollständigkeit kontrolliert, um fehlerhafte Bestellungen und Prozesse auszuschließen. Des Weiteren werden auch nicht-funktionale Anforderungen getestet, dazu zählt die Sicherheit, die Zuverlässigkeit und die Gebrauchstauglichkeit (Usability). Zusätzlich werden explizite Fehlertests durchgeführt. Diese enthalten absichtliche Fehler, um das Fehlermanagement des Systems auf seine Richtigkeit zu prüfen.

Abschließend wird vor Ort ein Abnahme- und Betriebstest durchgeführt, um einen reibungslosen Ablauf zu gewährleisten. Damit kann sichergestellt werden, dass die Software funktioniert und Ihren Wünschen entspricht. Die von Ihnen gestellten Anforderungen an die Software:

- die Zuverlässigkeit des Systems unter Last
- Verhinderung von Datenverlust und Abstürzen
- Schutz vor unabsichtlichen Veränderungen an Daten und Programmen
- EN ISO 9241 –11 Standarderfüllung
- Antwortzeit des Programms < 0,5s
- Wartbarkeit sowie Skalierbarkeit

PFLICHTENHEFT

12 Abkürzungsverzeichnis

Abkürzung	Beschreibung
CAP	Cloud Application Programming Model

Table 7: Abkürzungsverzeichnis