

# System Design



*Version:* 1.1

*Datum:* 17.11.2016

*Autoren:* Jana Dreiling, Marius Koch, Tina Brunner, Philip Waritschlager

## Änderungsverzeichnis:

Version	Datum	Autor(en)	Änderung
1.0	16.11.16	Alle	Setup
1.1	17.11.16	Alle	Zusammenstellung

# Inhaltsverzeichnis

<b>1 Einleitung .....</b>	<b>4</b>
1.1 Ziel dieses Dokuments .....	4
<b>2 Überblick .....</b>	<b>5</b>
<b>3 Funktionales Modell .....</b>	<b>6</b>
3.1 Freezer Account Management .....	6
3.1.1 UC_accountAnlegen .....	6
3.1.2 UC_accountLoeschen .....	7
3.1.3 UC_accountBearbeiten .....	8
3.2 Freezer Bestellungsmanagement.....	9
3.2.1 UC_einkaufszettelAnzeigen .....	9
3.2.2 UC_einkaufszettelProduktHinzufügen .....	10
3.2.3 UC_einkaufszettelProduktLoeschen .....	11
3.2.4 UC_autoBestellungKonfig .....	12
3.2.5 UC_autoBestellungAktivieren.....	13
3.2.6 UC_autoBestellungDeaktivieren .....	14
3.2.7 UC_autoBestellungAufgeben .....	15
3.2.8 UC_manuBestellungAufgeben .....	16
3.2.9 UC_erstelleWarenkorb.....	17
3.2.10 UC_übermittelBestellung.....	18
3.3 Freezer Rechnungsmanagement.....	19
3.3.1 UC_rechnungenAnzeigen .....	19
3.3.2 UC_rechnungAnzeigen .....	20
3.4 Freezer Bestandsverwaltung.....	21
3.4.1 UC_bestandProduktHinzufügen.....	21
3.4.2 UC_bestandProduktHinzufügen.....	22
3.4.3 UC_bestandAnzeigen .....	23
3.4.4 UC_warnungAblaufdatum .....	24
3.5 Supermarkt Management.....	25
3.5.1 UC_bestellungenEntgegennehmen .....	25
3.5.2 UC_rechnungErstellen .....	26
3.5.3 UC_rechnungÜbermitteln.....	27
<b>4 Dialoglandkarte.....</b>	<b>28</b>
<b>5 IT Architektur .....</b>	<b>31</b>
5.1 Kontextabgrenzung.....	31
5.2 Bausteinsicht .....	31
5.3 Laufzeitsicht.....	31
5.4 Verteilungssicht .....	32
<b>6 Schnittstellen und Integration.....</b>	<b>34</b>
6.1 Schnittstellenspezifikation .....	34
<b>7 Datenmodell .....</b>	<b>35</b>
7.1 Datenspeicherungskonzept.....	35
7.2 Physikalische Datenbankstruktur .....	35
<b>8 Testspezifikation .....</b>	<b>37</b>
8.1 Teststrategie.....	37

8.2 Testfälle .....	37
<b>9 Projektmanagement .....</b>	<b>40</b>
9.1 Projektplan.....	40
9.2 Aufgabenverteilung (Querschnittsaufgaben).....	40
<b>10 Bibliographie.....</b>	<b>41</b>

# 1 Einleitung

## 1.1 Ziel dieses Dokuments

In diesem Systemdesign soll spezifiziert werden, welche Softwaremodule in den kommenden Wochen vom Team "Freezers" erstellt werden.

Das Dokument ist strukturiert in 110 Kapitel.

In Kapitel 5 sind unterschiedliche technische Sichten auf die Software aufgelistet, während die ausführliche fachliche Beschreibung der Use Cases in Kapitel 3 erläutert wird.

Eine Aussicht über den weiteren Verlauf des Projektes ist in Kapitel 9.1 zu finden.

# 2 Überblick

Im vorliegenden Dokument wird eine Verwaltungssoftware für Kühlschränke beschrieben. Außerdem wird das Gegenstück, die Bestellungsannahme eines Supermarktes, entwickelt.

Mit dem "Freezer Management" kann jeder Kühlschrankbesitzer den Bestand seines Gerätes automatisch überwachen und auffüllen lassen.

Jedes Lebensmittel, das der Nutzer seinem Kühlschrank hinzufügt, wird von der Software erfasst. Es ist also möglich, den Bestand jederzeit online im Webbrowser anzeigen zu lassen. Hierzu wird davon ausgegangen, dass neue Produkte beim Hinzufügen oder Entnehmen automatisch gescannt werden und das Softwaresystem eine entsprechende Benachrichtigung enthält.

Die Nutzung des Kühlschranks ist in keiner Weise beeinträchtigt. Es kommen jedoch neue Features hinzu:

- Der Nutzer kann das Sortiment des Supermarktes einsehen. Für jede Produktart kann eine Mindestbestandsmenge eingestellt werden. Wird diese unterschritten, so wird das Produkt auf einer Merkliste vorgemerkt.
- Für jede Produktart kann der Nutzer eine Wunschmenge einstellen.
- In einstellbaren, regelmäßigen Intervallen generiert die Software hieraus eine Bestellung, die vom Nutzer noch manuell erweiterbar oder veränderbar ist. Die Menge der Produkte orientiert sich an den definierten Wunschmengen.
- Die Bestellung wird entweder automatisch oder nach Nutzeraufforderung an den Supermarkt gesendet. Der Liefertermin wird entweder automatisch oder nach Nutzereingabe mit dem Supermarkt vereinbart.
- Bei Eingang der Lieferung nimmt die Software die Produkte in den Bestand auf und der Nutzer kann den neuen Bestand seines Kühlschranks begutachten.
- Die Kosten der Bestellungen sind den Rechnungen zu entnehmen, die ebenfalls online einzusehen sind.

Diese Features können in einem Online-Tool konfiguriert und verwendet werden.

Zweck des Projektes ist ein mobiler Zugriff auf den Kühlschrank. Es ist möglich, von überall aus den Bestand einzusehen, Bestellungen abzuschicken oder Wunschmengen oder Mindestbestandsmengen zu verändern, ohne vor Ort sein zu müssen. Desweiteren erspart man dem Nutzer der Software den Gang zum Supermarkt und das manuelle Erstellen einer Bestellliste für alltägliche Lebensmittel. Außerdem geht der Prozess bargeldlos vonstatten.

Die Kühlschrank-Verwaltung interagiert mit der Schnittstelle des Supermarktes. Der Supermarkt gibt das Sortiment zur Einsicht aus, nimmt Bestellungen an, vereinbart Liefertermine, versendet Bestellungen und stellt Rechnungen aus.

## 3 Funktionales Modell

### 3.1 Freezer Account Management

#### 3.1.1 UC\_accountAnlegen

Use Case	Account anlegen		UC-ID	UC_01
			Version	0.1
Ziel im Kontext	Ein Benutzer ist angelegt und kann sich mit seinen Login Daten, welche per E-Mail geschickt werden, auf der Managementseite des Kühlschranks anmelden.			
Akteure				
	Primär	Freezer Admin		
	Sekundär	Freezer System, Mailserver		
Vorbedingung		Es muss ein Admin Account vorhanden sein.		
Auslösendes Ereignis		Neuer Nutzer gewünscht		
Ablaufschritte				
	Standard	1. Aufruf der Seite „newUser“ 2. Eingabe der Nutzerdaten 3. Klick auf Button „createUser“		
	Alternative	Anlegen des Users in der Datenbank		
Nachbedingung				
	bei Erfolg	1. User wird vom System in der Nutzerdatenbank eingetragen 2. Dem User wird eine E-Mail geschickt		
	bei Misserfolg	-		
Referenz				
Bearbeitungskommentar				

### 3.1.2 UC\_accountLoeschen

Use Case	Account löschen		UC-ID	UC_02
			Version	0.1
Ziel im Kontext	Ein angelegter Benutzer kann aus dem System gelöscht werden			
Akteure				
	Primär	Freezer Admin		
	Sekundär	Freezer System		
Vorbedingung	Es muss ein Admin und ein weiterer Account vorhanden sein.			
Auslösendes Ereignis	Nutzer soll gelöscht werden			
Ablaufschritte				
	Standard	1. Aufruf der Seite „userOverview“ 2. Auswahl des Users 3. Klick auf den Button „Delete User“ 4. Dialogfenster bestätigen		
	Alternative	Löschen des Users aus der Datenbank.		
Nachbedingung				
	bei Erfolg	1. Der User wird vom System aus der Datenbank gelöscht		
	bei Misserfolg	-		
Referenz				
Bearbeitungskommentar				

### 3.1.3 UC\_accountBearbeiten

Use Case	Account bearbeiten	UC-ID	UC_03
		Version	0.1
Ziel im Kontext	Die Daten eines angelegten Users können bearbeitet werden.		
Akteure			
	Primär	Freezer Admin	
	Sekundär	Freezer System	
Vorbedingung		Es muss ein Admin und ein weiterer Account vorhanden sein.	
Auslösendes Ereignis		Nutzerdaten sollen angepasst werden.	
Ablaufschritte			
	Standard	1. Aufruf der Seite „userOverview“ 2. Auswahl des Users 3. Klick auf den Button „Edit User“ 4. Anpassen der Daten 5. Klick auf den Button „Confirm“	
	Alternative	Bearbeiten des Users in der Datenbank	
Nachbedingung			
	bei Erfolg	1. Die Daten des Users werden in der Datenbank angepasst.	
	bei Misserfolg	-	
Referenz			
Bearbeitungskommentar			

## 3.2 Freezer Bestellungsmanagement

### 3.2.1 UC\_einkaufszettelAnzeigen

Use Case	Einkaufszettel anzeigen	UC-ID	UC_05
		Version	0.1
Ziel im Kontext	Es werden alle Produkte angezeigt, die auf dem Einkaufszettel stehen		
Akteure			
	Primär	Freezer User	
	Sekundär	Freezer System	
Vorbedingung		Es muss ein User angelegt sein.	
Auslösendes Ereignis			
Ablaufschritte			
	Standard	1. Aufruf der Seite „shoppingList“	
	Alternative		
Nachbedingung			
	bei Erfolg		
	bei Misserfolg		
Referenz			
Bearbeitungskommentar			



### 3.2.2 UC\_einkaufszettelProduktHinzufügen

Use Case	Produkt zum Einkaufszettel hinzufügen		UC-ID	UC_06
			Version	0.1
Ziel im Kontext	Es wird ein Produkt dem Einkaufszettel hinzugefügt			
Akteure				
	Primär	Freezer User		
	Sekundär	Freezer System		
Vorbedingung		Es muss ein User angelegt sein.		
Auslösendes Ereignis				
Ablaufschritte				
	Standard	1. Aufruf der Seite „shoppingList“ 2. Klick auf den Button „Add Product“ 3. Auswahl des Produkts aus der Liste 4. Klick auf den Button „Add Product“ 5. Stückzahl auf dem Einkaufszettel anpassen		
	Alternative			
Nachbedingung				
	bei Erfolg			
	bei Misserfolg			
Referenz				
Bearbeitungskommentar				

### 3.2.3 UC\_einkaufszettelProduktLoeschen

Use Case	Produkt vom Einkaufszettel löschen		UC-ID	UC_07
			Version	0.1
Ziel im Kontext	Ein Produkt wird vom Einkaufszettel gelöscht			
Akteure				
	Primär	Freezer User		
	Sekundär	Freezer System		
Vorbedingung	1. Es muss ein User angelegt sein. 2. Es muss ein Produkt auf dem Einkaufszettel stehen			
Auslösendes Ereignis				
Ablaufschritte				
	Standard	1. Aufruf der Seite „shoppingList“ 2. Klick auf den Button „Delete Product“ hinter dem entsprechenden Produkt. 3. Bestätigung des Dialogs		
	Alternative			
Nachbedingung				
	bei Erfolg			
	bei Misserfolg			
Referenz				
Bearbeitungskommentar				

### 3.2.4 UC\_autoBestellungKonfig

Use Case	Automatische Bestellung konfigurieren		UC-ID	UC_08
			Version	0.1
Ziel im Kontext	Bestellintervall und Lieferzeit für die automatische Bestellung wird festgelegt			
Akteure				
	Primär	Freezer User		
	Sekundär	Freezer System		
Vorbedingung		1. User muss angelegt sein		
Auslösendes Ereignis				
Ablaufschritte				
	Standard	1. Aufruf der Seite „autoOrder“ 2. Eintragen des Auslösetags 3. Eintragen des Bestellungsintervalls 4. Eintragen des Mindestbestellwertes		
	Alternative			
Nachbedingung				
	bei Erfolg			
	bei Misserfolg			
Referenz				
Bearbeitungskommentar				

### 3.2.5 UC\_autoBestellungAktivieren

Use Case	Automatische Bestellung aktivieren		UC-ID	UC_09
			Version	0.1
Ziel im Kontext	Die automatische Bestellung wird aktiviert			
Akteure				
	Primär	Freezer User		
	Sekundär	Freezer System		
Vorbedingung	1. UC_autoBestellungKonfig 2. Automatische Bestellung ist deaktiviert			
Auslösendes Ereignis	-			
Ablaufschritte				
	Standard	1. Aufruf der Seite „autoOrder“ 2. Klick auf Button „aktivieren“		
	Alternative			
Nachbedingung				
	bei Erfolg	UC_autoBestellung		
	bei Misserfolg			
	Referenz			
Bearbeitungskommentar				

### 3.2.6 UC\_autoBestellungDeaktivieren

Use Case	Automatische Bestellung deaktivieren		UC-ID	UC_10
			Version	0.1
Ziel im Kontext	Die automatische Bestellung wird deaktiviert			
Akteure				
	Primär	Freezer User		
	Sekundär	Freezer System		
Vorbedingung	1. UC_autoBestellungKonfig 2. Automatische Bestellung ist aktiviert			
Auslösendes Ereignis	-			
Ablaufschritte				
	Standard	1. Aufruf der Seite „autoOrder“ 2. Klick auf Button „deaktivieren“		
	Alternative			
Nachbedingung				
	bei Erfolg			
	bei Misserfolg			
Referenz				
Bearbeitungskommentar				

### 3.2.7 UC\_autoBestellungAufgeben

Use Case	Automatisch Bestellung aufgeben	UC-ID	UC_11
		Version	0.1
Ziel im Kontext	Eine Bestellung wird automatisch erstellt und an den Supermarkt übermittelt		
Akteure			
	Primär	Freezer System	
	Sekundär		
Vorbedingung	1. UC_autoBestellungKonfig 2. UC_autoBestellungAktiviert		
Auslösendes Ereignis	Bestelltag wurde nach Intervall erreicht		
Ablaufschritte			
	Standard	1. UC_erstelleWarenkorb 2. Vergleiche Mindestbestellwert mit Bestellwert a. Wenn erreicht: UC_übermittelBestellung b. Wenn nicht erreicht: Annulliere Bestellung + schicke Warnungs E-Mail an User.	
	Alternative		
Nachbedingung			
	bei Erfolg		
	bei Misserfolg	Nächste Ausführung erst nach Ende des nächsten Intervalls	
Referenz			
Bearbeitungskommentar			

### 3.2.8 UC\_manuBestellungAufgeben

Use Case	Manuelle Bestellung aufgeben	UC-ID	UC_12
		Version	0.1
Ziel im Kontext	Eine Bestellung wird manuell erstellt und an den Supermarkt übermittelt		
Akteure			
	Primär	Freezer User	
	Sekundär	Freezer System	
Vorbedingung	1. User ist angelegt 2. Der Einkaufszettel enthält Produkte		
Auslösendes Ereignis			
Ablaufschritte			
	Standard	1. Seit „shoppingList“ aufrufen 2. Button „Create Order“ drücken 3. UC_erstelleWarenkorb 4. Übersicht über die Bestellung wird angezeigt 5. Button „Submit Order“ drücken	
	Alternative		
Nachbedingung			
	bei Erfolg		
	bei Misserfolg		
Referenz			
Bearbeitungskommentar			

### 3.2.9 UC\_erstelleWarenkorb

Use Case	Erstelle Warenkorb		UC-ID	UC_13
			Version	0.1
Ziel im Kontext	Der Warenkorb wird auf Basis der Einkaufsliste zusammen gestellt			
Akteure				
	Primär	Freezer System		
	Sekundär			
Vorbedingung	1. Ein User ist angelegt 2. Bestellung wird ausgelöst			
Auslösendes Ereignis				
Ablaufschritte				
	Standard	1. Produkte auf dem Einkaufszettel werden mit dem Bestand verglichen 2. Nicht vorhandene Produkte und Produkte mit einer Stückzahl unter dem Mindestbestand werden auf die Bestellliste gesetzt.		
	Alternative			
Nachbedingung				
	bei Erfolg			
	bei Misserfolg			
Referenz				
Bearbeitungskommentar				



### 3.2.10 UC\_übermittelBestellung

Use Case		UC-ID	UC_14
		Version	0.1
Ziel im Kontext	Die Artikel des Warenkorbs werden als Bestellung an den Supermarkt übermittelt		
Akteure			
	Primär	Freezer System	
	Sekundär		
Vorbedingung	1. UC_erstelleWarenkorb 2. Bestellung aufgeben a. UC_autoBestellungAufgeben b. UC_manuBestellungAufgeben		
Auslösendes Ereignis			
Ablaufschritte			
	Standard	Bestellung wird an den Supermarkt übermittelt	
	Alternative		
Nachbedingung			
	bei Erfolg		
	bei Misserfolg		
Referenz			
Bearbeitungskommentar			

### 3.3 Freezer Rechnungsmanagement

#### 3.3.1 UC\_rechnungenAnzeigen

Use Case	Rechnungen Anzeigen	UC-ID	UC_15
		Version	0.1
Ziel im Kontext	Dem User wird eine Liste aller Rechnungen angezeigt		
Akteure			
	Primär	Freezer User	
	Sekundär	Freezer System	
Vorbedingung		User ist angelegt	
Auslösendes Ereignis			
Ablaufschritte			
	Standard	Aufruf der Seite „invoices“	
	Alternative		
Nachbedingung			
	bei Erfolg		
	bei Misserfolg		
Referenz			
Bearbeitungskommentar			

### 3.3.2 UC\_rechnungenAnzeigen

Use Case	Rechnungen Anzeigen	UC-ID	UC_16
		Version	0.1
Ziel im Kontext	Dem User wird eine Liste aller Rechnungen angezeigt		
Akteure			
	Primär	Freezer User	
	Sekundär	Freezer System	
Vorbedingung	1. UC_rechnungenAnzeigen 2. Eine Rechnung ist vorhanden		
Auslösendes Ereignis			
Ablaufschritte			
	Standard	Klick auf eine Rechnung	
	Alternative		
Nachbedingung			
	bei Erfolg		
	bei Misserfolg		
Referenz			
Bearbeitungskommentar			

### 3.4 Freezer Bestandsverwaltung

#### 3.4.1 UC\_bestandProduktHinzufügen

Use Case	Produkt zur Bestandsliste hinzufügen	UC-ID	UC_17
		Version	0.1
Ziel im Kontext	Der Bestandsliste des Kühlschranks wird ein Produkt hinzugefügt		
Akteure			
	Primär	Freezer System	
	Sekundär		
Vorbedingung			
Auslösendes Ereignis		Ein Produkt wird in den Kühlschrank gestellt	
Ablaufschritte			
	Standard	1. Das reingestellte Produkt wird in die Bestandsliste hinzugefügt	
	Alternative		
Nachbedingung			
	bei Erfolg		
	bei Misserfolg		
Referenz			
Bearbeitungskommentar			

### 3.4.2 UC\_bestandProduktHinzufügen

Use Case	Produkt von Bestandsliste entfernen	UC-ID	UC_18
		Version	0.1
Ziel im Kontext	Von der Bestandsliste des Kühlschranks wird ein Produkt entfernt		
Akteure			
	Primär	Freezer System	
	Sekundär		
Vorbedingung			
Auslösendes Ereignis		Ein Produkt wird aus dem Kühlschrank entnommen	
Ablaufschritte			
	Standard	1. Das entnommene Produkt wird aus der Bestandsliste entfernt	
	Alternative		
Nachbedingung			
	bei Erfolg		
	bei Misserfolg		
Referenz			
Bearbeitungskommentar			

### 3.4.3 UC\_bestandAnzeigen

Use Case	Bestandsliste anzeigen	UC-ID	UC_19
		Version	0.1
Ziel im Kontext	Die Bestandsliste des Kühlschranks wird angezeigt		
Akteure			
	Primär	Freezer User	
	Sekundär	Freezer System	
Vorbedingung		Ein User ist angelegt	
Auslösendes Ereignis			
Ablaufschritte			
	Standard	1. Aufruf der Seite „inventory“	
	Alternative		
Nachbedingung			
	bei Erfolg		
	bei Misserfolg		
Referenz			
Bearbeitungskommentar			

### 3.4.4 UC\_warnungAblaufdatum

Use Case	Warnen bei ablaufendem Produktdatum		UC-ID	UC_20
			Version	0.1
Ziel im Kontext	Dem User wird eine E-Mail geschickt, sobald ein Produkt kurz vor dem Ablaufdatum in der Bestandsliste ist.			
Akteure				
	Primär	Freezer System Freezer Mail		
	Sekundär			
Vorbedingung	1. Ein User ist angelegt 2. Ein Produkt kurz vor dem Ablaufdatum ist in der Bestandsliste			
Auslösendes Ereignis				
Ablaufschritte				
	Standard	1. Dem User wird eine Benachrichtigung geschickt		
	Alternative			
Nachbedingung				
	bei Erfolg			
	bei Misserfolg			
Referenz				
Bearbeitungskommentar				

### 3.5 Supermarkt Management

#### 3.5.1 UC\_bestellungenEntgegennehmen

Use Case		UC-ID	UC_21
		Version	0.1
Ziel im Kontext	Der Supermarkt empfängt die Bestellung des Kunden		
Akteure			
	Primär	Supermarkt System	
	Sekundär		
Vorbedingung		UC_übermittelBestellung	
Auslösendes Ereignis			
Ablaufschritte			
	Standard	1. Die Bestellung wird gespeichert 2. Lieferung beauftragen	
	Alternative		
Nachbedingung			
	bei Erfolg	UC_rechnungErstellen	
	bei Misserfolg		
Referenz			
Bearbeitungskommentar			



### 3.5.2 UC\_rechnungErstellen

Use Case	Rechnung erstellen	UC-ID	UC_22
		Version	0.1
Ziel im Kontext	Das System erstellt die Rechnung auf Basis der Bestellung		
Akteure			
	Primär	Freezer User	
	Sekundär	Freezer System	
Vorbedingung		UC_bestellungEntgegennehmen	
Auslösendes Ereignis			
Ablaufschritte			
	Standard	Die Rechnung wird erstellt	
	Alternative		
Nachbedingung			
	bei Erfolg	UC_rechnungÜbermitteln	
	bei Misserfolg		
Referenz			
Bearbeitungskommentar			

### 3.5.3 UC\_rechnungÜbermitteln

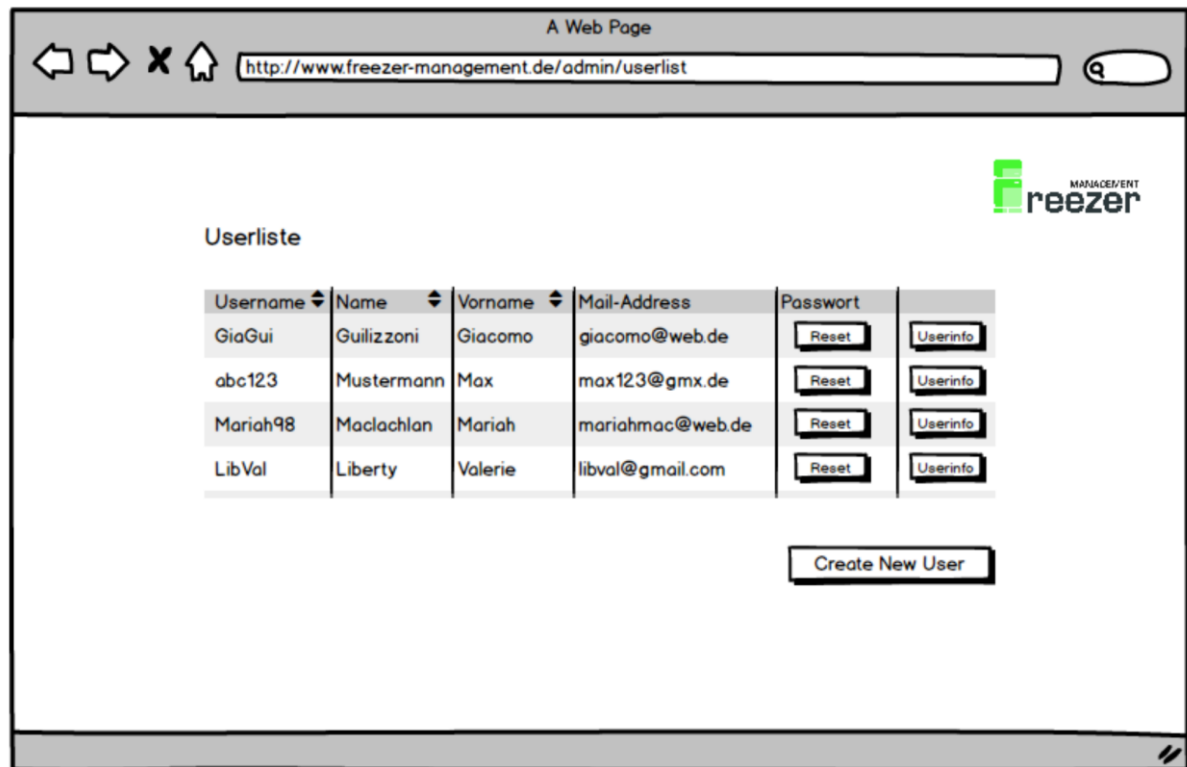
Use Case	Rechnung übermitteln		UC-ID	UC_23
			Version	0.1
Ziel im Kontext	Das System übermittelt die Rechnung an das Kühlschranksystem			
Akteure				
	Primär	Supermarkt System		
	Sekundär			
Vorbedingung	UC_bestellungErstellen			
Auslösendes Ereignis				
Ablaufschritte				
	Standard	Rechnung wird an den Kunden übermittelt		
	Alternative			
Nachbedingung				
	bei Erfolg			
	bei Misserfolg			
Referenz				
Bearbeitungskommentar				

## 4 Dialoglandkarte

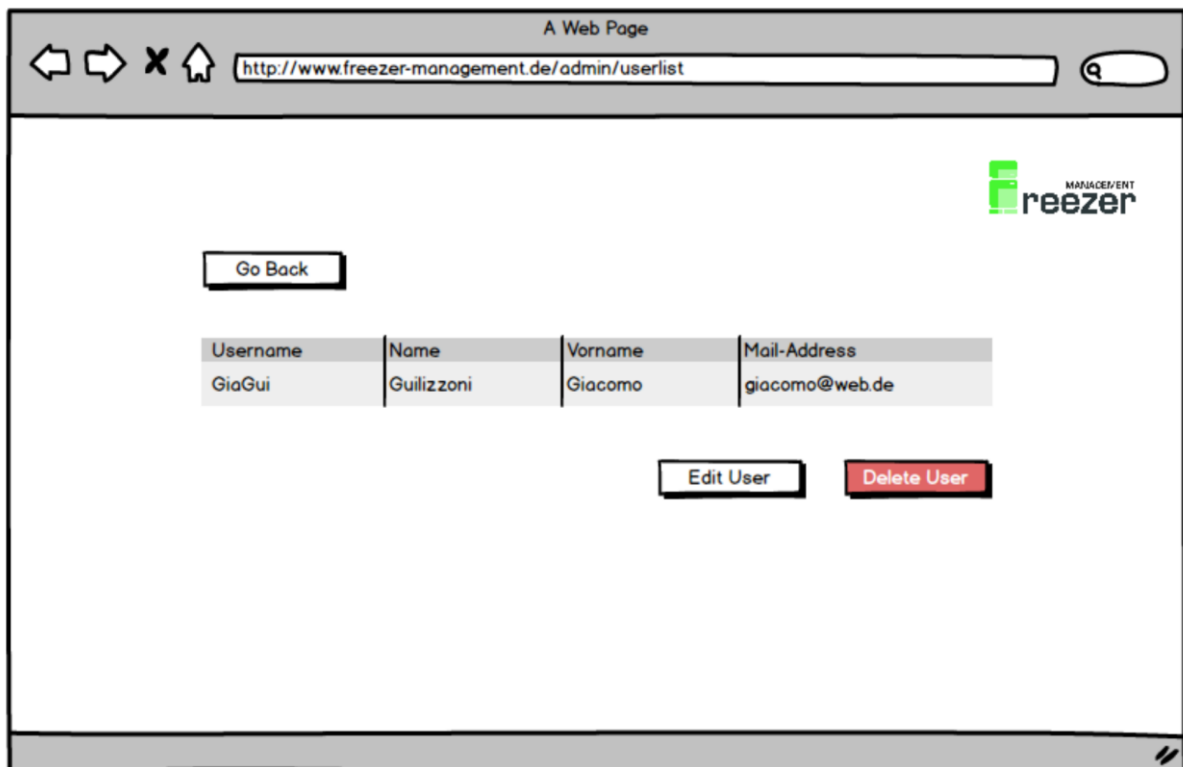
Ein interaktives Mockup ist unter Kap. 10 verlinkt.

### Administrative Dialoge:

Zeigt dem Admin die Liste aller User. Der Admin kann das Passwort zurücksetzen und die Info zu einem User aufrufen. Außerdem kann er einen neuen User erzeugen.



Der Admin kann einen einzelnen User einsehen und editieren, bzw. löschen.



Der Admin kann einen neuen User erzeugen.

The screenshot shows a web browser window titled 'A Web Page' with the address bar displaying 'http://www.freezer-management.de/admin/userlist'. The page content includes the 'reezer' logo in the top right corner. The main heading is 'Create a new User'. Below this, there are four input fields labeled 'Surname', 'Name', 'Username', and 'Mail-Address'. At the bottom right of the form area, there are two buttons: a red 'Cancel' button and a white 'Save' button.

A Web Page

http://www.freezer-management.de/admin/userlist

reezer

Create a new User

Surname

Name

Username

Mail-Address

Cancel Save

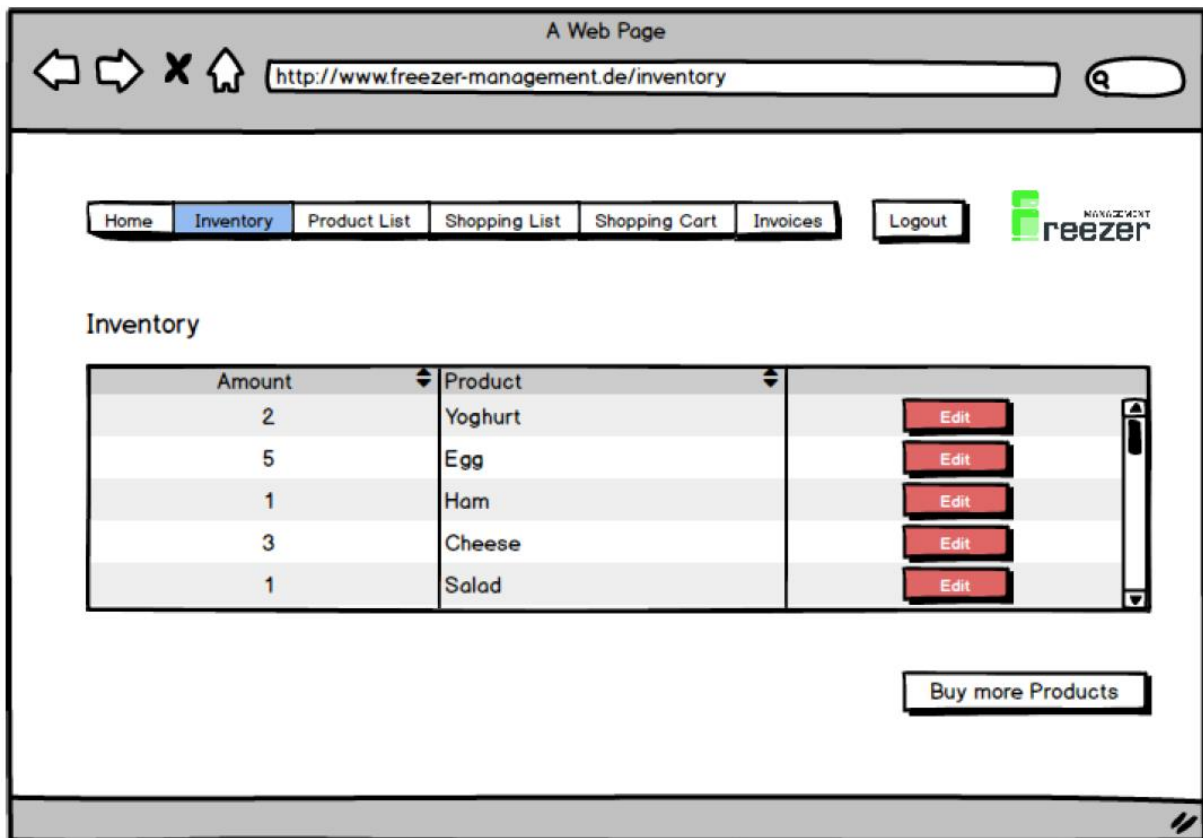
Diese Oberfläche stellt den Login zur Freezerverwaltung dar. Username und Passwort werden mit den Einträgen aus der Datenbank verglichen und bei Erfolg kann sich der Nutzer einloggen.

A hand-drawn sketch of a web browser window. The title bar says "A Web Page". The address bar contains "http://www.freezer-management.de". The page content includes the "reezer" logo (a green square icon followed by the text "reezer" and "MANAGER/ENT" above it) in the top right corner. In the center, there are two input fields: the first is labeled "Username" and the second is labeled "Passwort". Below the "Passwort" field is a button labeled "Login".

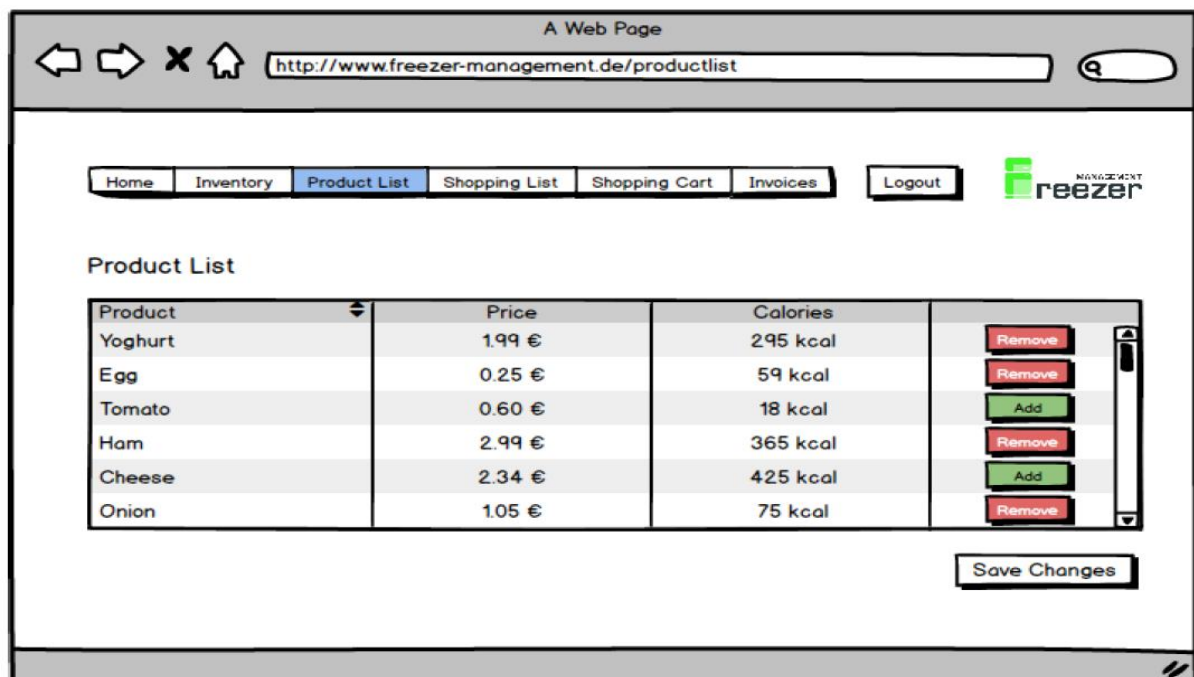
Hier werden alle möglichen Verwaltungsoptionen angezeigt. Durch den Aufruf einer der Optionen gelangt der Nutzer in die jeweilige Verwaltung.

A hand-drawn sketch of a web browser window. The title bar says "A Web Page". The address bar contains "http://www.freezer-management.de/home". The page content includes the "reezer" logo in the top right corner. In the center, there are seven rounded rectangular buttons arranged in two rows. The top row contains four buttons: "Inventory", "Product List", "Shopping List", and "Shopping Cart". The bottom row contains three buttons: "Invoices", "My Account", and "Logout".

Hier wird die Bestandsliste dargestellt. Anhand dieser kann der Nutzer sehen, welche Produkte in seinem Kühlschrank vorhanden sind und sie nach Wunsch editieren (z.B. die Stückzahl bearbeiten). Wahlweise kann man durch den Klick auf den „Buy more Products“ auf die Supermarktverwaltung gelangen, in der weitere Produkte bestellt werden können.



Diese Abbildung stellt die Produktliste dar, die alle Produkte aus der Supermarktverwaltung enthält. Nach Bedarf kann der Nutzer ein Produkt zu seiner Bestellliste hinzufügen bzw. entfernen.



Hier wird die Bestellliste dargestellt. Diese beinhaltet alle Produkte, die der Nutzer entweder regelmäßig oder einmalig bestellen will. Anhand eines bestimmten Wochentages und Bestellintervalls wird demnach eine Bestellung ausgelöst, sollte ein regelmäßiges Produkt den Mindestbestellwert erreicht haben.

Des Weiteren hat der Nutzer auch die Möglichkeit per „Create Order“ eine manuelle Bestellung zu aktivieren, sollte er die Waren sofort brauchen, anstatt auf die automatische Bestellung zu warten.

Product	Minimum Stock	Maximum Stock	regularly	Delete
Yoghurt	2	5	<input checked="" type="checkbox"/>	Delete
Eggs	5	15	<input checked="" type="checkbox"/>	Delete
Ham	-	3	<input type="checkbox"/>	Delete
Cheese	1	4	<input checked="" type="checkbox"/>	Delete
Milk	1	3	<input checked="" type="checkbox"/>	Delete
Salad	-	2	<input type="checkbox"/>	Delete

Add Product Create Order

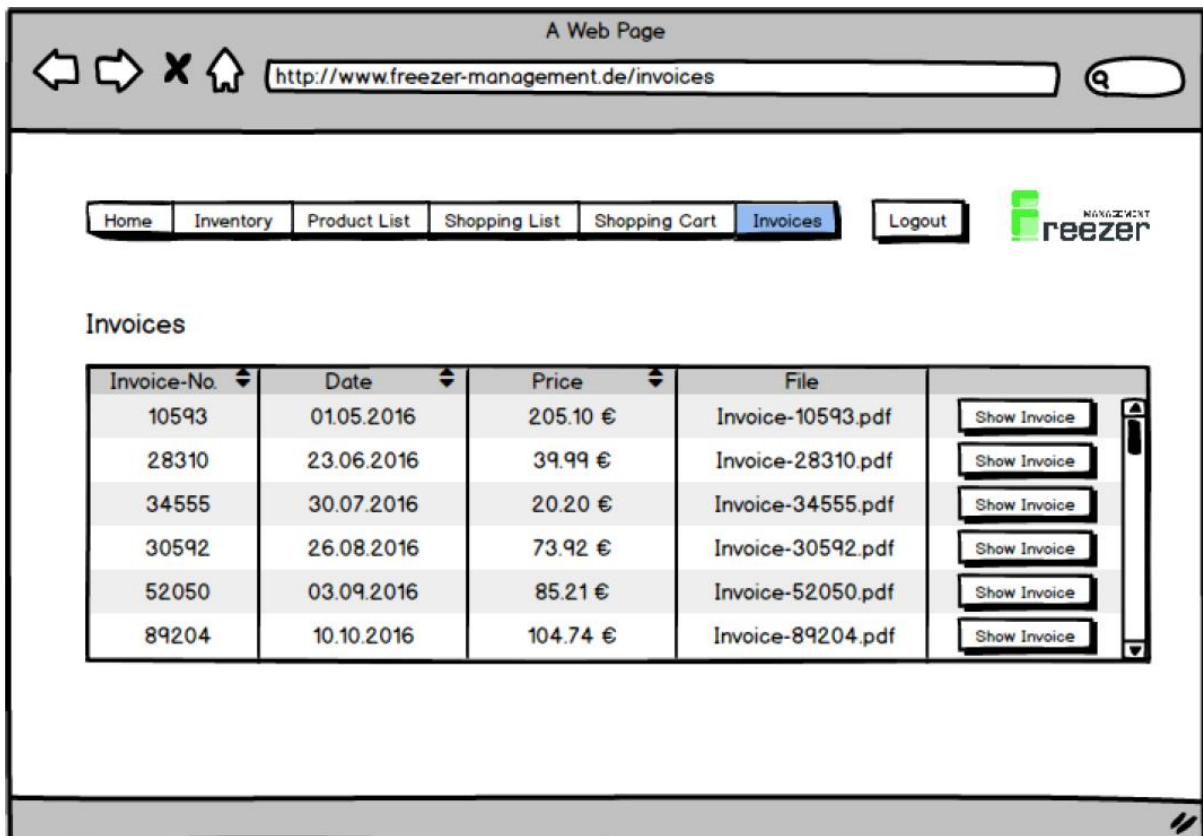
Hier wird der Einkaufskorb dargestellt. Will der Nutzer ein bestimmtes Produkt nicht bestellen, hat er die Möglichkeit das jeweilige Produkt hier abzuwählen und anschließend die Bestellung abzuschicken.

Anzahl	Product	Price per Piece	Sum Price	Remove
3	Yoghurt	1.99 €	5.97 €	Remove
5	Egg	0.25 €	1.25 €	Remove
1	Tomato	0.60 €	0.60 €	Remove
2	Ham	2.99 €	5.98 €	Remove
1	Cheese	2.34 €	2.34 €	Remove
5	Onion	1.05 €	5.25 €	Remove

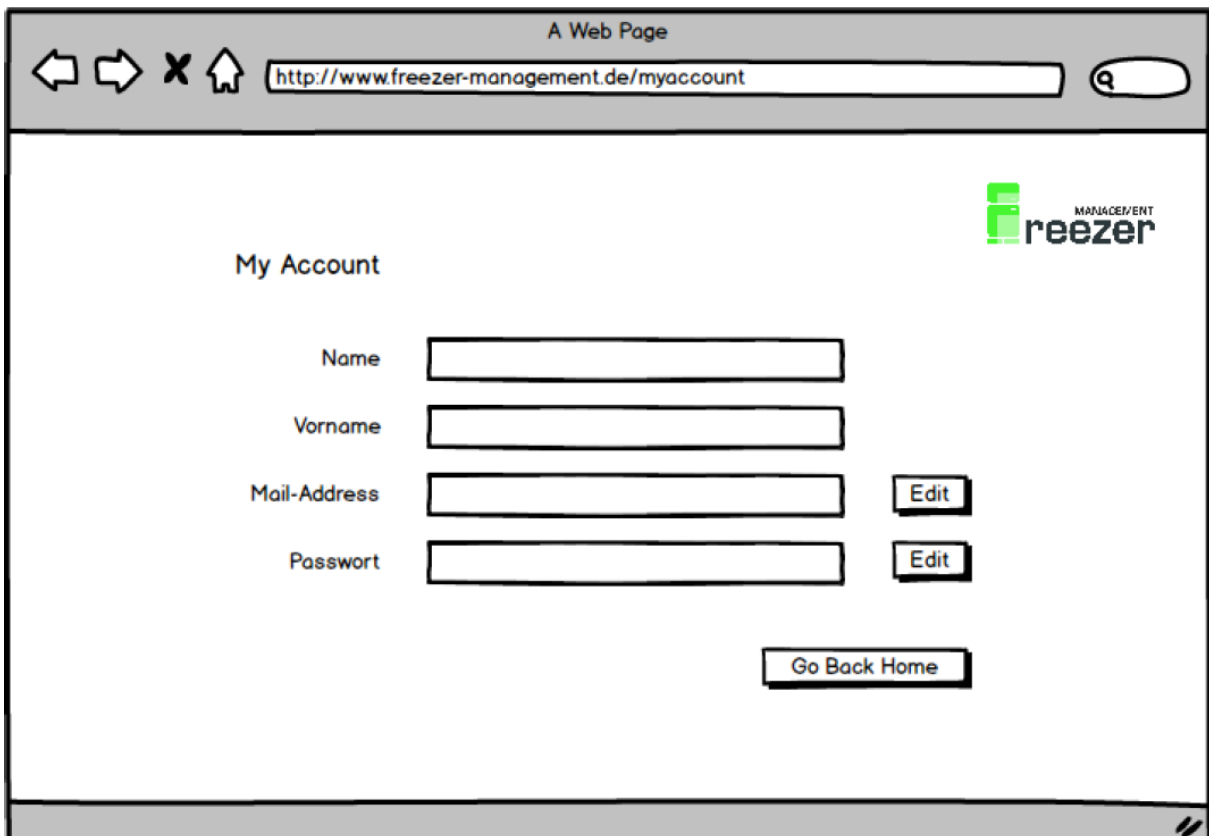
Submit Order



Hier wird die Übersicht aller Bestellungen angezeigt. Diese wird mit Nummer, Datum der Bestellung, Gesamtpreis und der Bestellung als PDF gespeichert. Letztere lässt sich per „Show Invoice“ anzeigen.



Diese Maske stellt den Account des Kunden dar. Hier können die jeweiligen Daten überarbeitet und gespeichert werden.



Diese Oberfläche dient zur Bestätigung der Änderung der E-Mail Adresse.

A Web Page

http://www.freezer-management.de/myaccount/edit-mailaddress

reezer MANAGEMENT

### Change Mail-Address

old Mail-Address

new Mail-Address

repeat new Mail-Address

This is a hand-drawn style screenshot of a web browser window. The title bar says 'A Web Page'. The address bar contains 'http://www.freezer-management.de/myaccount/edit-mailaddress'. The page content includes the 'reezer' logo with 'MANAGEMENT' in small text above it. The main heading is 'Change Mail-Address'. Below it are three text input fields labeled 'old Mail-Address', 'new Mail-Address', and 'repeat new Mail-Address'. At the bottom are two buttons: 'Cancel' and 'Save'.

Diese Oberfläche dient zur Bestätigung der Änderung des Anmeldepassworts.

A Web Page

http://www.freezer-management.de/myaccount/edit-password

reezer MANAGEMENT

### Change Password

old Password

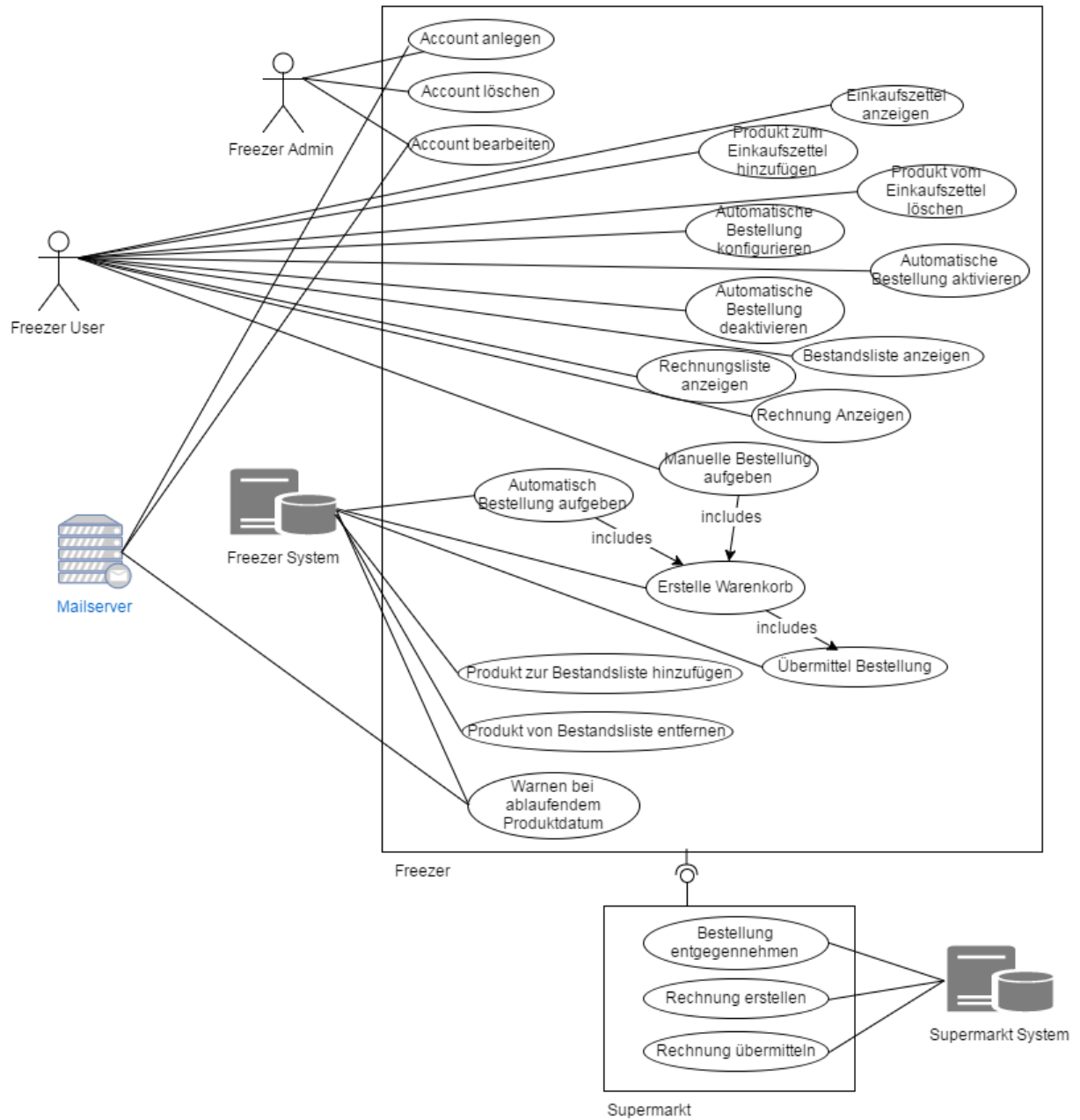
new Password

repeat new Password

This is a hand-drawn style screenshot of a web browser window. The title bar says 'A Web Page'. The address bar contains 'http://www.freezer-management.de/myaccount/edit-password'. The page content includes the 'reezer' logo with 'MANAGEMENT' in small text above it. The main heading is 'Change Password'. Below it are three text input fields labeled 'old Password', 'new Password', and 'repeat new Password'. At the bottom are two buttons: 'Cancel' and 'Save'.

## 5 IT Architektur

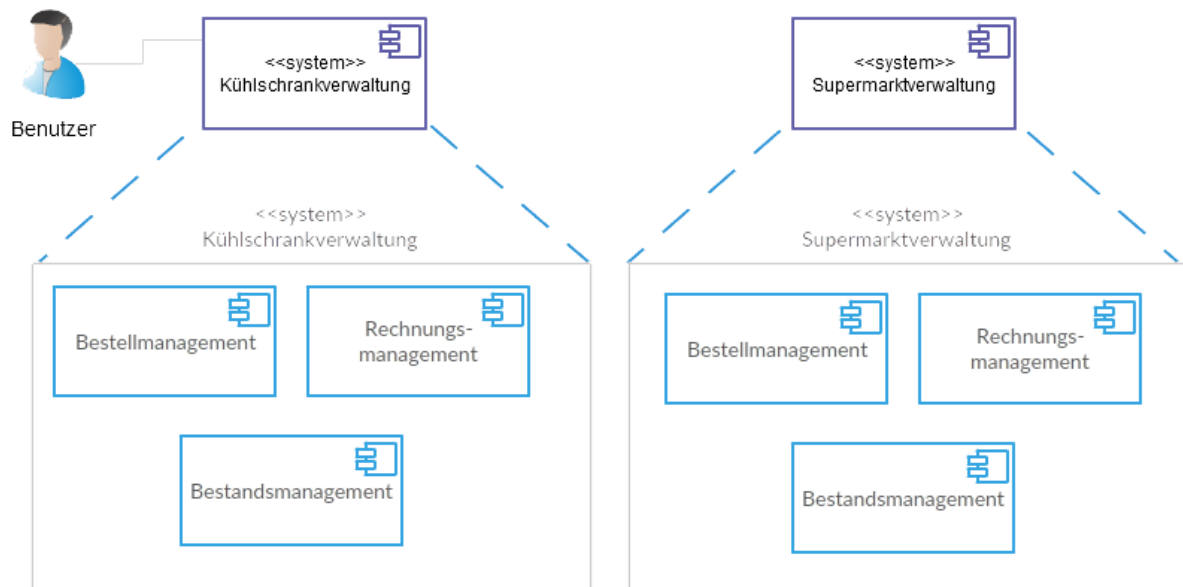
### 5.1 Kontextabgrenzung



## 5.2 Bausteinsicht

Der Benutzer hat nur Zugriff auf das System der Kühlschrankverwaltung. Diese besteht jeweils aus einem Bestell-, Rechnungs- und Bestandsmanagement. Die Supermarktverwaltung läuft auf einem separaten System und besitzt dieselbe innere Struktur wie die Kühlschrankverwaltung, jedoch haben die einzelnen Module eine andere Funktionsweise.

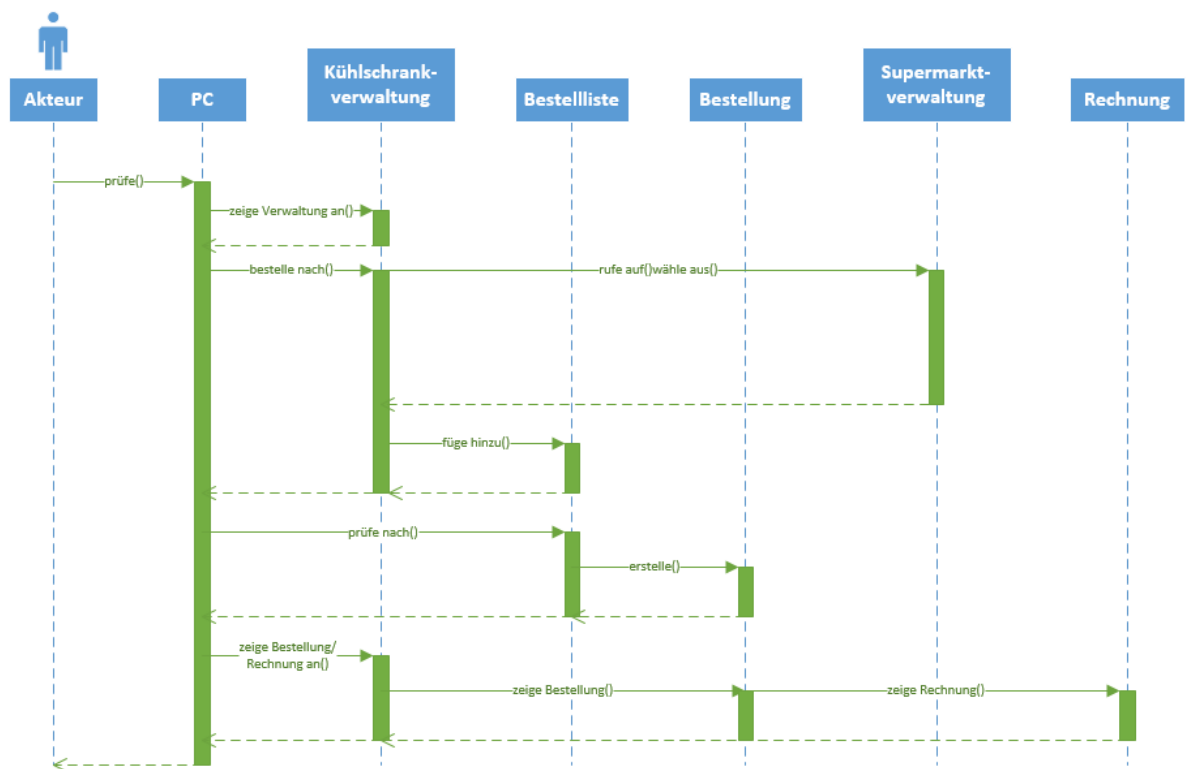
So wird z.B. in der Kühlschrankverwaltung beim Bestellmanagement eine Liste zu bestellender Artikel verwaltet, während die Supermarktverwaltung eine daraus folgende Bestellung verarbeiten und umsetzen muss.



### 5.3 Laufzeitsicht

Folgende Abfolge ist bei einem herkömmlichen Gebrauch des Freezermanagements zu erwarten:

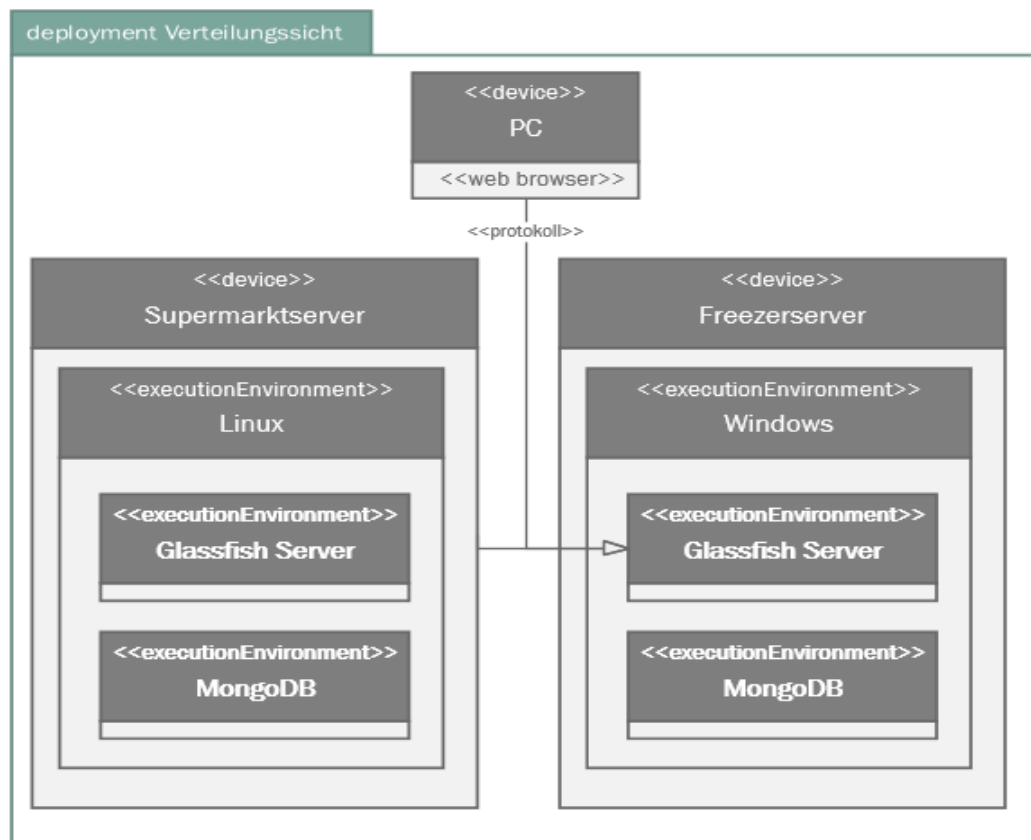
1. Der Akteur greift über einen PC auf die Kühlschranksverwaltung zu. In dieser erhält er eine Übersicht aller wichtigen Daten, wie z.B. den aktuellen Bestand, die nächste fällige Bestellung etc.
2. Der Akteur entscheidet sich dafür, eine bestimmte Ware zu bestellen. Über die Kühlschranksverwaltung gelangt er auf die Supermarktverwaltung und wählt das jeweilige Produkt aus. Dieses wird anschließend auf die Bestellliste hinzugefügt.
3. Ab einem bestimmten Fälligkeitsdatum, an der die Bestellliste geprüft wird, wird je nachdem, ob diese etwas enthält, eine Bestellung ausgelöst.
4. Abschließend kann sich der Akteur über die Kühlschranksverwaltung die Bestellungen bzw. Rechnungen anzeigen lassen.



## 5.4 Verteilungssicht

Die Verteilungssicht besteht aus hauptsächlich 3 Komponenten.

1. Die Anwendung PC, die durch einen Web Browser über JSON auf den Server der Kühlschranksverwaltung zugreifen kann (Freezerserver)
2. Der Supermarktserver, welcher auf Basis von Linux auf einem Glassfish Server läuft, während die Datenbank auf MongoDB basiert. Dieser greift über JSON auf den Freezerserver zu.
3. Der Freezerserver, welcher auf Basis von Windows auf einem Glassfish Server läuft und dessen Datenbank ebenfalls auf MongoDB basiert.



## 6 Schnittstellen und Integration

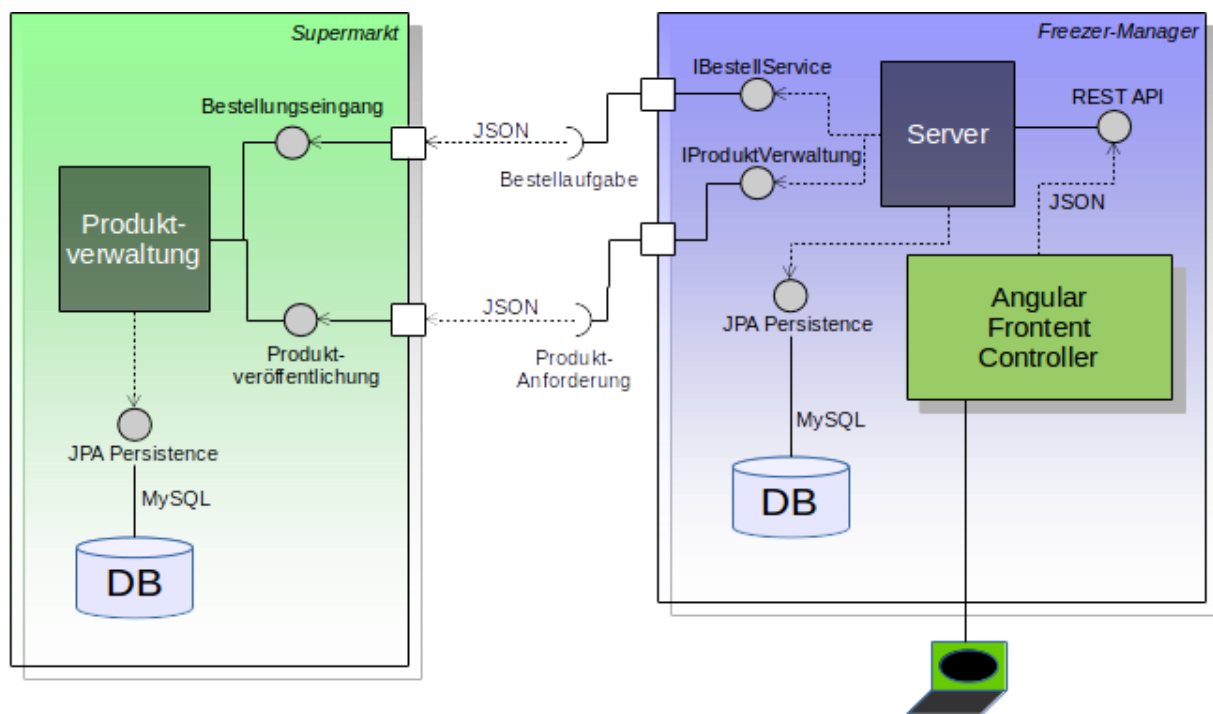
### 6.1 Schnittstellenspezifikation

Systemübergreifende Schnittstellen

- Asynchron
  - Das Sortiment des Supermarktes wird vom Freezer Management unidirektional als JSON-Objekt abgerufen. (Produktveröffentlichung)
  - Bestellungen werden zwischen Freezer Management und Supermarkt ausgemacht. Hier ist bidirektionaler Datenaustausch im JSON-Format gegeben. Der Bestellservice bzw. die Produktverwaltung des Servers interagiert also indirekt mit dem Bestellungseingang bzw. der Produktveröffentlichung des Supermarktes.

Komponentenübergreifende Schnittstellen

- Asynchron
  - Das Frontend Framework greift auf die Daten zu, die im Freezer Management hinterlegt sind. Hier wird eine RESTful API verwendet, generiert durch JavaEE, genutzt durch AngularJS. Der Bestand wird als JSON-Objekt dargestellt.
- Synchron
  - Freezer Management Server sowie Supermarkt holen sich die momentanen Daten im Laufe eines Requests mittels JDBC aus ihrer jeweiligen Datenbank.



## 7 Datenmodell

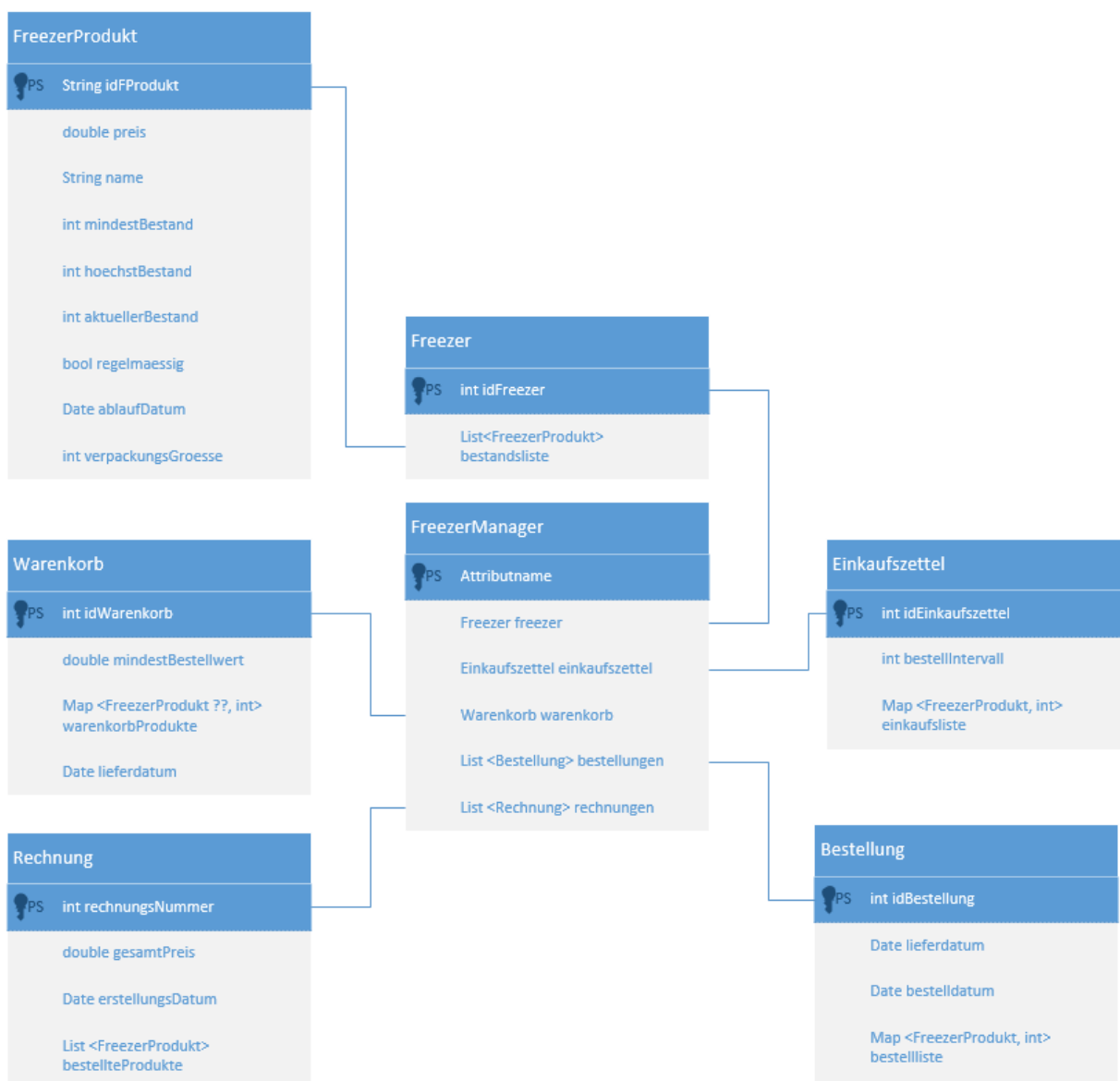
### 7.1 Datenspeicherungskonzept

Beide Systeme (Freezer Management und Supermarkt) speichern alle ihrer Daten, die speicherrelevant sind, in eine MongoDB MySQL Datenbank.

### 7.2 Physikalische Datenbankstruktur

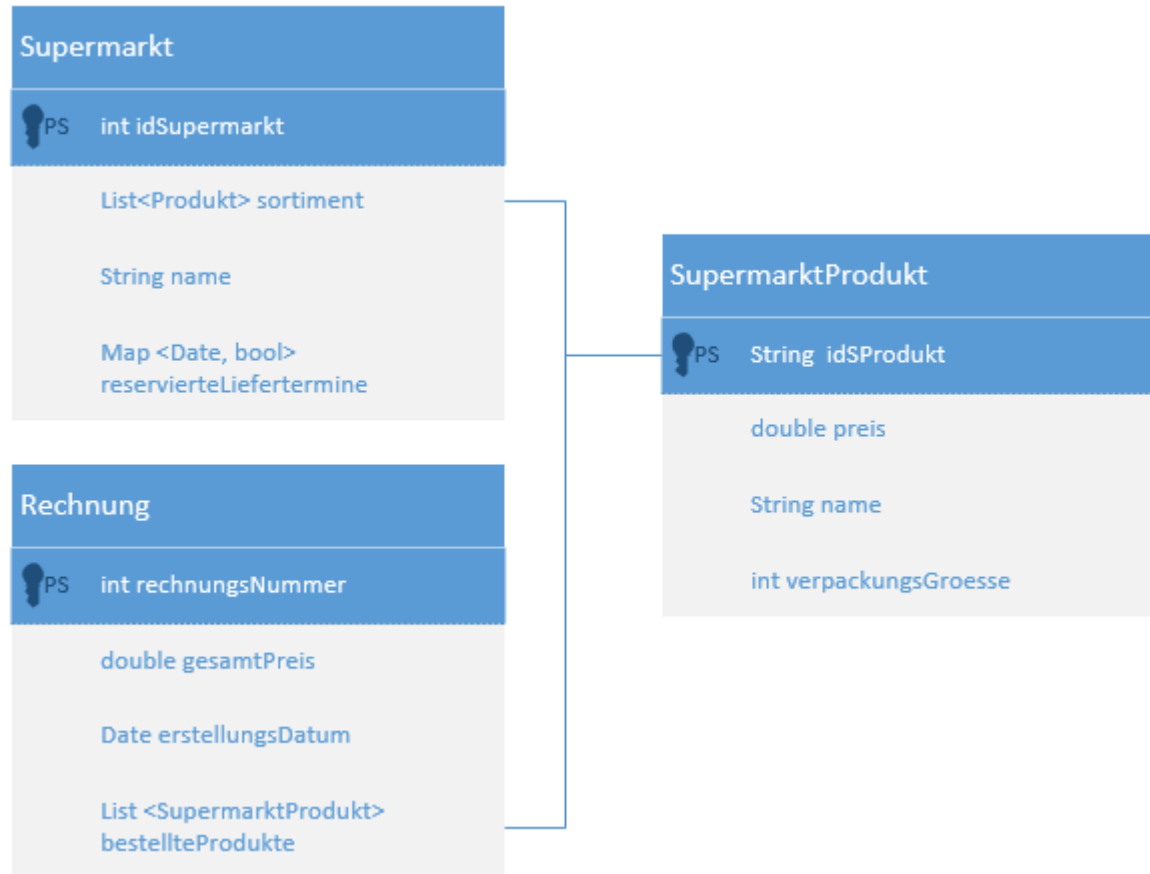
Datenmodell des Freezer Managements:

Der Freezer merkt sich die zentrale Information Mindest- und Wunsch/Höchstbestand in der Tabelle "FreezerProdukt". Die Produkte, die vom Nutzer extra gewünscht sind, sind in "Einkaufszettel" zu finden.





Datenmodell des Supermarktes:  
Der Supermarkt verwaltet ein Sortiment und vereinbart Lieferungen.



## 8 Testspezifikation

### 8.1 Teststrategie

- Frontend
  - AngularJS controller werden mittels eines Frameworks wie Protractor (Jasmine) halb automatisiert getestet.  
Mit manuellen Tests wird die Nutzerfreundlichkeit der Website überprüft
- Backend
  - Die JavaEE Funktionalität wird mit Unit-Tests sichergestellt, halb automatisierte Integrationstests (z.B. Ejb3Unit) stellen die korrekte Kommunikation der Systeme fest.

### 8.2 Testfälle

- Freezer Management
  - Account anlegen
    - Administrator erstellt neuen Nutzer
    - Ein neuer Account muss in der Datenbank angelegt sein
  - Account löschen
    - Administrator nutzt die User-Löschfunktion
    - Eintrag muss aus der Datenbank gelöscht werden
  - Account bearbeiten
    - Administrator nutzt die User-Bearbeiten-Funktion
    - Eintrag muss in der Datenbank entsprechend modifiziert werden
  - Neues Produkt
    - Nutzer fügt ein Produkt dem Kühlschrank hinzu
    - Beim nächsten Aufruf der Bestandsliste muss der aktuelle Stand zu sehen sein
  - Bestand ausgeben
    - Nutzer wählt "Bestand anzeigen"
    - Kompletter und korrekter Bestand muss ausgegeben werden
  - Sortiment ausgeben
    - Nutzer wählt "Supermarkt Sortiment anzeigen"
    - Freezer Management muss bei Supermarkt Sortiment anfragen
    - Supermarkt muss vollständiges Sortiment zurückgeben
    - Alle Produkttypen des Supermarktes müssen vollständig in der Oberfläche angezeigt werden
  - Produkt der Bestellliste hinzufügen
    - Nutzer fügt Produkt der Bestellliste hinzu

- Datenbank muss neuen Eintrag mit entsprechender Produkttyp-ID und Anzahl enthalten
- Produkt von der Bestellliste löschen
  - Nutzer löscht Produkt von der Bestellliste
  - Datenbankeintrag muss gelöscht werden
- Automatische Bestellung konfigurieren
  - Nutzer stellt präferierten Intervall für die Produktmengencheck ein
  - Datenbankeintrag des Intervalls muss aktualisiert werden
- Automatische Bestellung aktivieren
  - Nutzer aktiviert automatische Bestellung
  - Datenbankeintrag automatischeBestellung wird auf 1 gesetzt
- Automatische Bestellung deaktivieren
  - Nutzer deaktiviert automatische Bestellung
  - Datenbankeintrag automatischeBestellung wird auf 0 gesetzt
- Mindest- und Wunschbestandsmenge eines Produkttyps angeben
  - Nutzer ist in Sortimentanzeige und wählt einen momentan nicht gewünschten Produkttyp, oder  
Nutzer ist in Bestandsliste und wählt einen vorhandenen Produkttyp
  - Nutzer konfiguriert eine Mindestbestandsmenge und eine Wunschbestandsmenge des Produkttyps
  - Mindest- und Wunschbestandsmenge muss umgehend am Server hinterlegt werden
- Mindestbestandsmenge wird unterschritten
  - Nutzer entnimmt so viele Produkte eines Produkttyps, dass gilt:  
 $\text{Mindestbestandsmenge} < \text{Bestandsmenge}$
  - Dynamisch generierte Bestellliste muss Differenz aus Wunschmenge und Bestandsmenge des jeweiligen Produkttyps enthalten
- Bestellung modifizieren
  - Eine Bestellung wird generiert und dem Nutzer bekannt gegeben (Mail)
  - Nutzer verringert oder vergrößert Produktmengen oder fügt der Bestellung weitere Produkte hinzu
  - Bestellung, die am Server ankommt, muss modifizierte Produktliste enthalten
- Terminabsprache
  - Freezer Management schickt eine Bestellung ab
  - Supermarkt muss eine Liste an möglichen Lieferterminen zurücksenden
  - Nutzer wählt einen Liefertermin im Webbrowser
  - Supermarkt muss Liefertermin bestätigen
- Bestellungseingang
  - Supermarkt erhält Bestellung
  - Produktlieferung muss eingeleitet werden

- Rechnung muss erstellt und versendet werden
- Rechnungseingang
- Bestellung wird versendet
- Betrag der Bestellung muss dem Betrag der Rechnung entsprechen
- Projektmanagement

### 8.3 Projektplan

Kommende Termine:

- **22.12.2016** Zwischenbesprechung (Implementierungsstand)
- **12.01.2016** Abgabe
- **19.01.2017** Präsentation / mündliche Prüfung

Wöchentlich donnerstags kurze Besprechungen des Fortschritts mit Hr. Kroll

### 8.4 Aufgabenverteilung (Querschnittsaufgaben)

Tina Brunner

Datenbank-Design

Jana Dreiling

Backend Logik

Marius Koch

Schnittstellen

Philip Waritschlager

ng-GUI

## 9 Biographie

- [https://www.dropbox.com/s/z9u9v56vikd5r7w/Mockups\\_FreezerManagement.zip?dl=0](https://www.dropbox.com/s/z9u9v56vikd5r7w/Mockups_FreezerManagement.zip?dl=0)