

TECHNISCHE UNIVERSITÄT DRESDEN
SOFTWAREPROJEKT GRUPPE 30
GEEK-SHOP

Entwicklerdokumentation

Sebastian Döring, Felix Döring, Marcus Kammerdiener, Dominik Lauck, Elizaveta Ragozina

<http://is63050.inf.tu-dresden.de/~swt14w30/index.php>

Inhaltsverzeichnis

1 Einführung und Ziele	2
1.1 Aufgabenstellung	2
1.2 Stakeholder	3
2 Randbedingungen	3
2.1 Technische Randbedingungen	3
2.2 Konventionen	3
3 Kontextabgrenzung	4
3.1 Fachlicher Kontext	4
3.2 Externe Schnittstellen	5
4 Bausteinsicht	7
5 Laufzeitsicht	8
5.1 Persistenz	14
5.2 Benutzungsoberfläche	15
5.3 Ergonomie	15
5.4 Transaktionsbehandlung	15
5.5 Sessionbehandlung und Sicherheit	15
5.6 Plausibilisierung und Validierung	16
5.7 Konfigurierbarkeit	16
5.8 Internationalisierung	16
5.9 Buildmanagement	16

1 Einführung und Ziele

1.1 Aufgabenstellung

„Think Nerd“ – The Shop for your nerdy needs.

Ein echter Computer-Nerd hat ganz besondere Bedürfnisse. Wir wissen das selbst am besten. Daher haben wir uns entschlossen, einen Shop zu eröffnen, der von einfachen koffeinreichen Minzpastillen über das monatliche Linuxmagazin bis hin zur Armbanduhr mit WLAN-Empfänger alles Wichtige anbietet. „Think Nerd“ ist dabei nicht nur ein Name, sondern vielmehr unsere Geschäftspraxis. Schnell war klar, wir wollen ein eigens für uns entwickeltes Verwaltungssystem. Das Programm soll primär an der Kasse eingesetzt werden und den Verkaufsvorgang elektronisch abwickeln können. Bei uns gibt es nur normale Angestellte und den Ladenbesitzer.

Um das Geschäftsklima zu verbessern, hätten wir gern, dass jeder Angestellte mit einem zufälligem Nerd-Witz begrüßt wird. Damit sich die Witze nicht ständig wiederholen, soll der Ladenbesitzer die Möglichkeit haben, Witze entsprechend zu verwalten.

Einzelne Mitarbeiter sollen sich mit ihrem Namen und einem sicheren Passwort anmelden können. Unsichere Passwörter sollen bei uns gar nicht erst zugelassen werden. Ein Passwort wie „mama“ wäre also unerwünscht. Möglich wäre z. B. „MaM4.12\$“.

Mitarbeiter sollen vor allem Einkäufe von Kunden abwickeln. Unser Sortiment ist in Kategorien wie z. B. „Nerd-Wear“ oder „Elektronische Gadgets“ gegliedert. Die Kategorien sind ihrerseits wieder in Unterkategorien wie z. B. „Admin-Shirts“ o. „Gamer-Shirts“ unterteilt. Mitarbeiter sollen sich zunächst durch das Sortiment „klicken“ können oder über eine Suchfunktion Waren direkt finden. Angezeigte Ergebnisse sollen nach allen Kriterien (auf/absteigend nach Preis, nach Name ...) sortiert werden können, wie man es von gängigen Online-Shops kennt. Die Angestellten sollen die Möglichkeit haben, die Waren nacheinander in einen Warenkorb einzufügen. Im nächsten Schritt sollen die Angestellten eine Übersicht über den Einkauf erhalten und die Zahlungsweise eingeben können. Es ist für Kunden möglich per elektronischem Lastschriftverfahren, per Kreditkarte oder bar zu zahlen. Nach der Bezahlung soll eine druckfertige Rechnung angezeigt werden.

Wir haben eine 14-tägige Geld-zurück-Garantie. Falls ein Kunde etwas reklamieren möchte, soll(en) der/die Artikel ebenfalls in einen Warenkorb gelegt werden. Im nächsten Schritt soll noch einmal die Übersicht über die Reklamation gezeigt werden. Es ist dann erforderlich, dass der Abschluss einer Reklamation vom Ladenbesitzer genehmigt werden muss. Am besten wäre es, wenn er dazu Name und Passwort eingeben müsste.

Der Ladenbesitzer ist für die Verwaltung der Mitarbeiter zuständig. Er kann sie entlassen, einstellen oder ihre persönlichen Daten verändern. Nur der Ladenbesitzer kann das Sortiment bearbeiten. Das heißt er kann Kategorien und Unterkategorien erstellen, löschen und umbenennen. Ebenfalls kann nur er Artikel hinzufügen, löschen und bearbeiten.

Ein Überblick über den aktuellen Lagerbestand ist ebenfalls erforderlich. Der Ladenbesitzer muss den Bestand natürlich auch verändern können, um neue Lieferungen einzutragen. Wenn der Bestand eines Artikels eine untere Grenze unterschreitet, möchte der Ladenbe-

sitzer natürlich gewarnt werden.

Der Ladenbesitzer kann auf „Rohdaten“ aller Verkäufe zugreifen. Also wann, welcher Artikel, in welcher Menge verkauft wurde. Wir planen bei einem Freelancer ein umfangreiches Statistik-Tool für uns in Auftrag zu geben. Damit dieser gleich diese Rohdaten benutzen kann, brauchen wir eine Exportfunktion, die die Daten in eine XML Datei schreibt.

1.2 Stakeholder

Rolle	Beschreibung	Ziel/Intention	Bemerkungen
Owner	Ladenbesitzer	GELD	alle Rechte
Employee	Mitarbeiter	Arbeiten → Geld	Sklave

2 Randbedingungen

2.1 Technische Randbedingungen

Hardware-Vorgaben	RAM: 128 MB Datenträgerkapazität: 150 MB mindestens 266MHz-Prozessor
Software-Vorgaben	Java 8 Browser mit HTML5-Unterstützung
Programmiervorgaben	Java 8 Thymeleaf Maven 3.2.5 Spring-Framework 4.1.4 SalesPoint-Framework 6.1.1

2.2 Konventionen

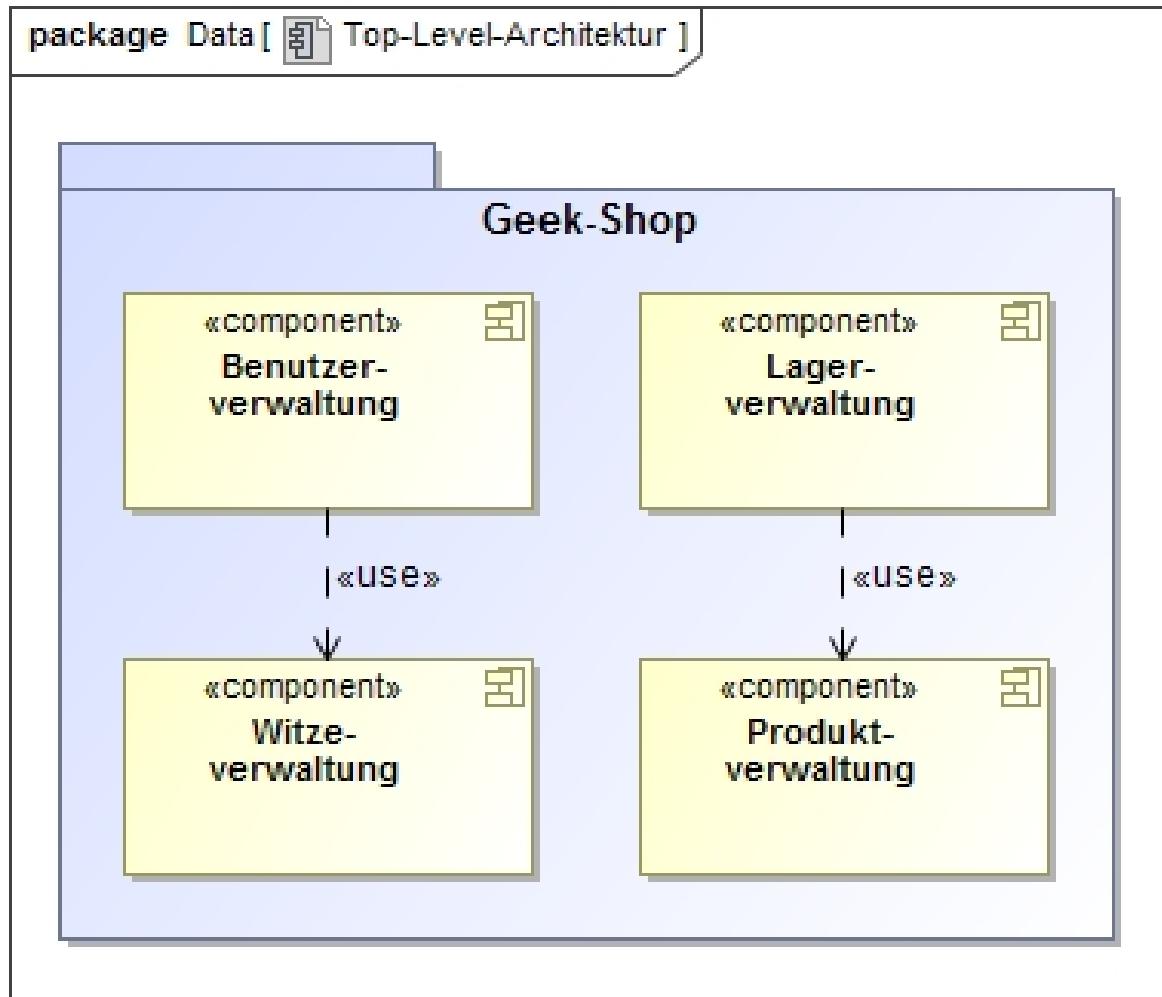
Zur optischen Unifizierung des Codes und zur Optimierung der *Imports* wurde die Autoformatierung von IntelliJ IDEA 14 verwendet.

Die Namenskonvention lautet wie folgt:

- Klassennamen beginnen mit einem Großbuchstaben, bestehen sie aus mehreren Wörtern, so werden sie zusammen geschrieben und jeweils der erste Buchstabe groß
- Methodennamen beginnen mit einem Kleinbuchstaben, bestehen sie aus mehreren Wörtern, so werden sie zusammen und ab dem zweiten Wort die Anfangsbuchstaben groß geschrieben.

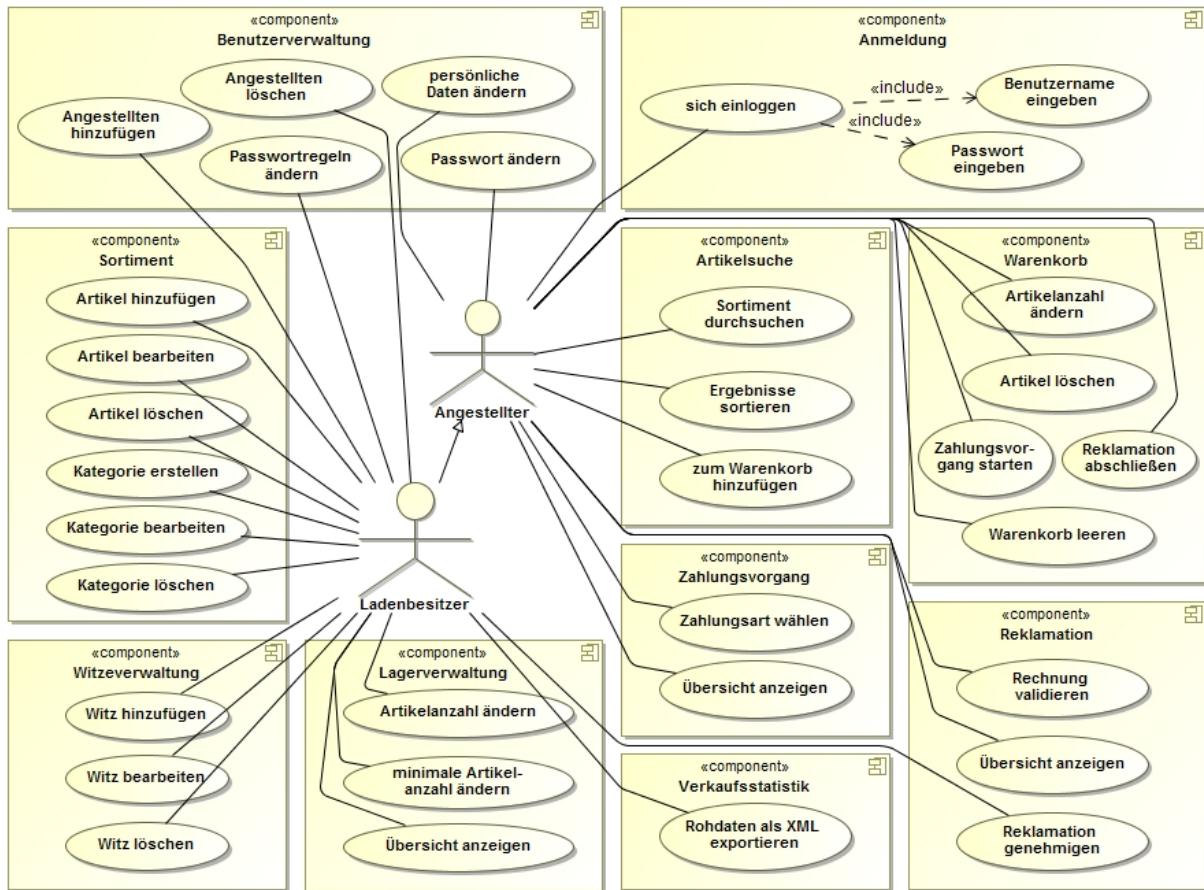
3 Kontextabgrenzung

3.1 Fachlicher Kontext



Die Top-Level-Architektur besteht aus der Benutzerverwaltung, welche die Witzeverwaltung nutzt und der Lagerverwaltung, welche auf die Produktverwaltung zugreift.

3.2 Externe Schnittstellen

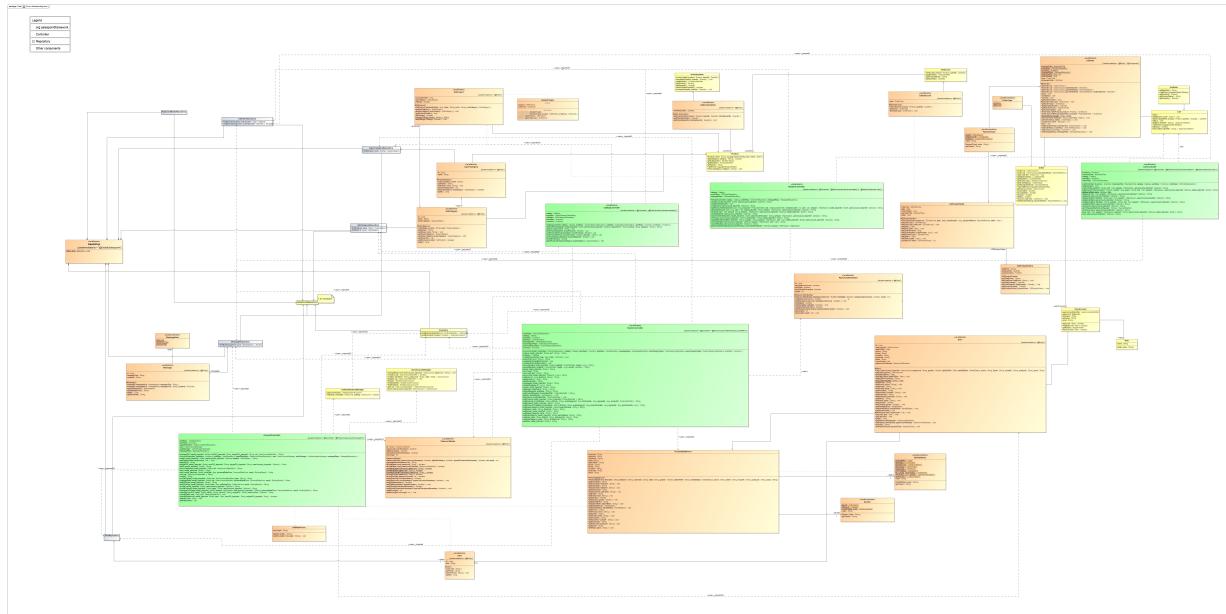


Anwendungsfall	Beschreibung	Akteur
Benutzerverwaltung		
Angestellten hinzufügen	Fügt einen neuen Angestellten hinzu	Ladenbesitzer
Angestellten löschen	Löscht einen vorhandenen Angestellten	Ladenbesitzer
persönliche Daten ändern	Ändert die persönlichen Daten des Nutzers	Angestellter
Passwortregeln ändern	Ändert die Regeln der Passwortvalidierung	Ladenbesitzer
Passwort ändern	Ändert das Passwort des Nutzers	Angestellter
Anmeldung		
sich einloggen	Loggt den Nutzer ein	Angestellter

Sortiment		
Artikel hinzufügen	Fügt einen Artikel zum Sortiment hinzu	Ladenbesitzer
Artikel bearbeiten	Verändert einen vorhandenen Artikel im Sortiment	Ladenbesitzer
Artikel löschen	Löscht einen vorhandenen Artikel aus dem Sortiment	Ladenbesitzer
Kategorie erstellen	Fügt eine neue Kategorie hinzu	Ladenbesitzer
Kategorie bearbeiten	Verändert eine vorhandene Kategorie	Ladenbesitzer
Kategorie löschen	Löscht eine vorhandene Kategorie	Ladenbesitzer
Artikelsuche		
Sortiment durchsuchen	Sucht Artikel aus dem Sortiment	Angestellter
Ergebnisse sortieren	Sortiert die gefundenen Artikel wie gewünscht	Angestellter
zum Warenkorb hinzufügen	Fügt Artikel zum Warenkorb hinzu	Angestellter
Warenkorb		
Artikelanzahl ändern	Ändert die Anzahl des Artikels im Warenkorb	Angestellter
Artikel löschen	Entfernt einen Artikel aus dem Warenkorb	Angestellter
Zahlungsvorgang starten	Beginnt den Zahlungsvorgang	Angestellter
Reklamation abschließen	Erstellt offene Reklamation	Angestellter
Warenkorb leeren	Löscht alle Artikel aus dem Warenkorb	Angestellter
Witzeverwaltung		
Witz hinzufügen	Fügt einen Witz hinzu	Ladenbesitzer
Witz bearbeiten	Verändert einen vorhandenen Witz	Ladenbesitzer
Witz löschen	Löscht einen vorhandenen Witz	Ladenbesitzer

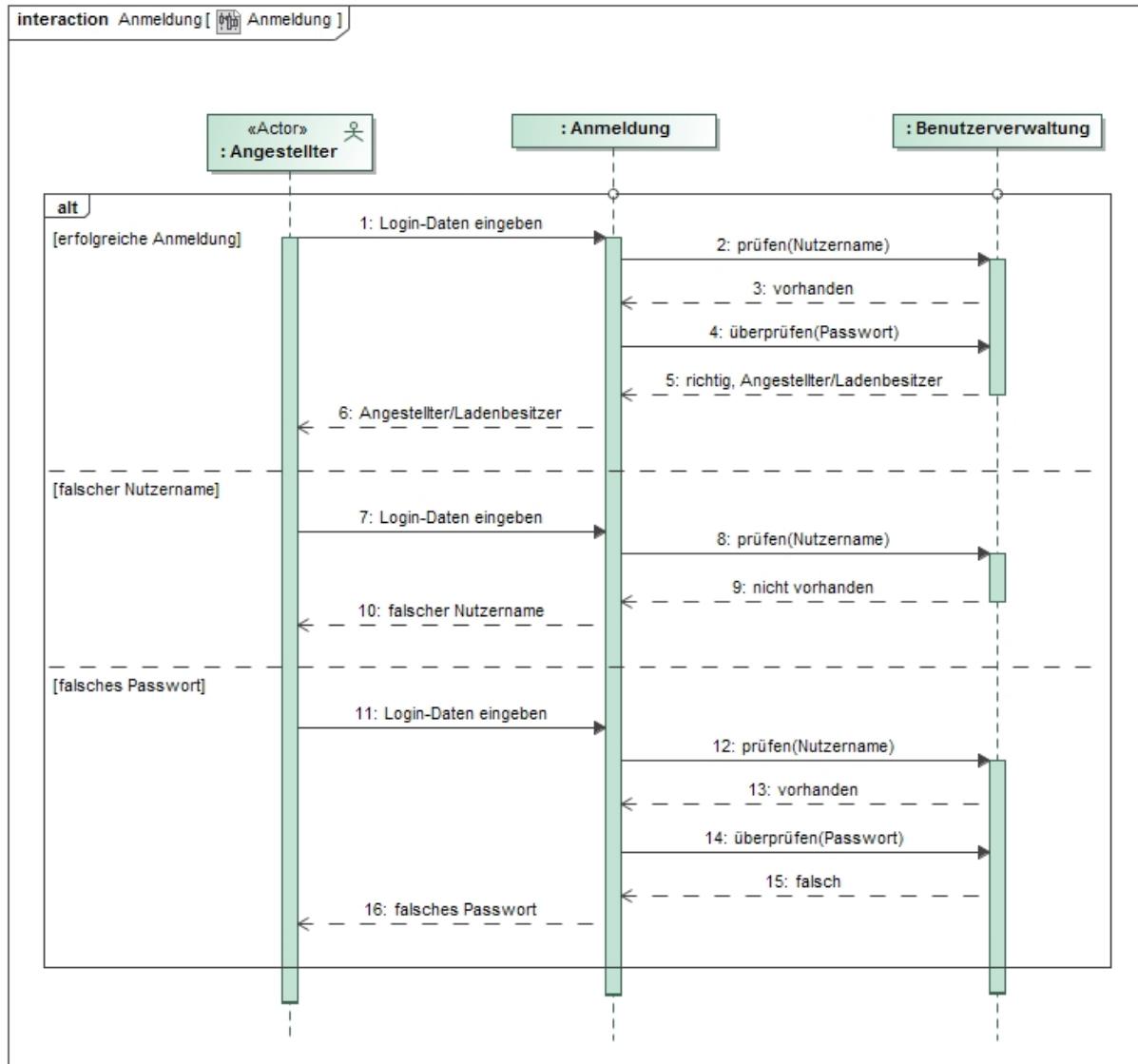
Lagerverwaltung		
Artikelanzahl änder	Ändert die Anzahl des Artikels im Bestand	Ladenbesitzer
minimale Artikelanzahl änder	Ändert die minimale Anzahl des Artikels im Bestand	Ladenbesitzer
Übersicht anzeigen	Zeigt eine Übersicht über alle vorhandenen Artikel	Ladenbesitzer
Zahlungsvorgang		
Zahlungsart auswählen	Wählt die Zahlungsart aus	Angestellter
Übersicht anzeigen	Zeigt die Bestellübersicht an	Angestellter
Verkaufsstatistik		
Rohdaten als XML exportieren	Exportiert die Verkaufsstatistik als XML	Ladenbesitzer
Reklamation		
Rechnung validieren	Überprüft, ob die Rechnung noch im Zeitrahmen liegt	Angestellter
Übersicht anzeigen	Zeigt alle Produkte der Rechnung an	Angestellter
Reklamation genehmigen	Genehmigt die Reklamation und schließt sie ab	Ladenbesitzer

4 Bausteinsicht



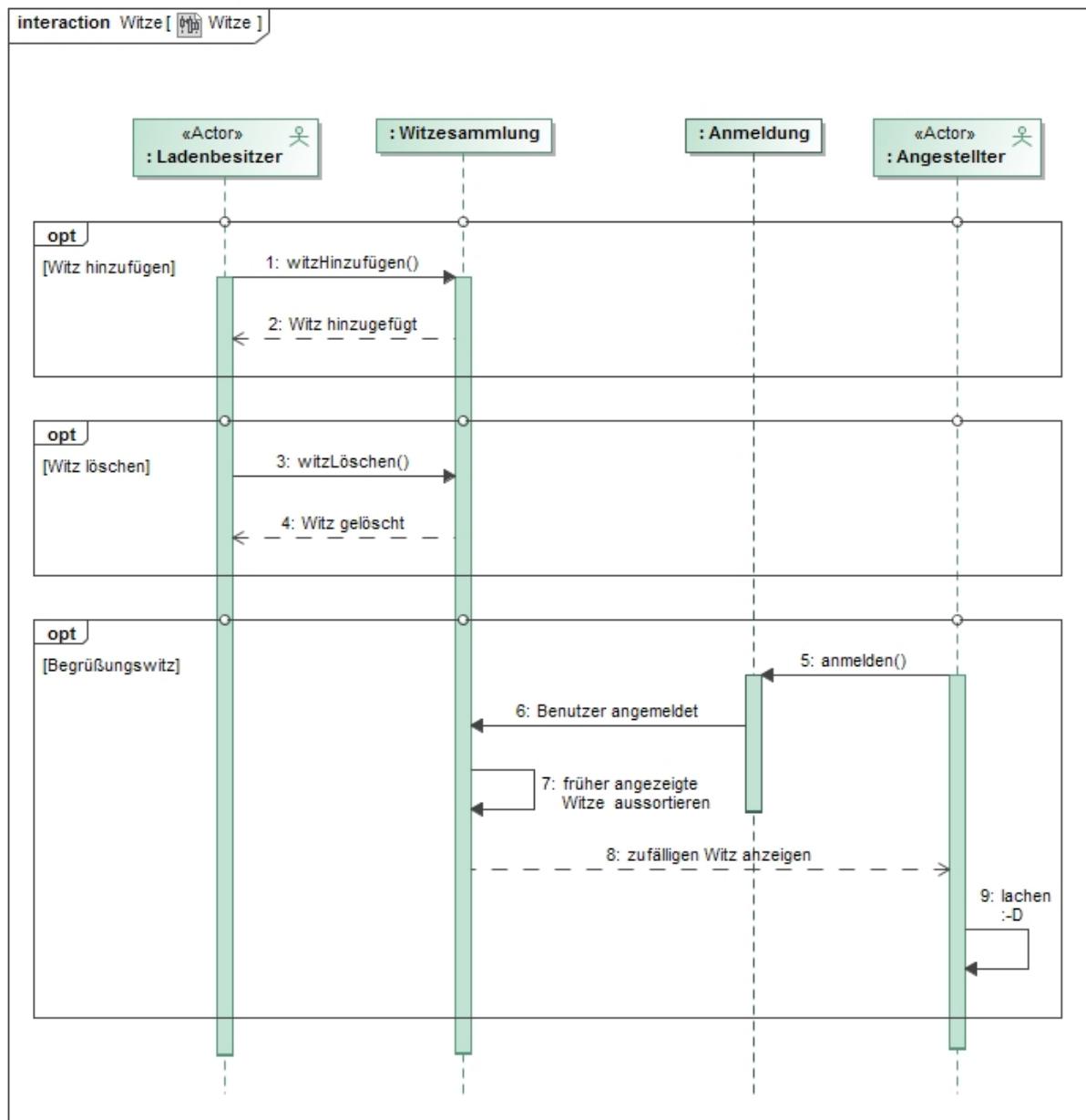
5 Laufzeitsicht

Anmeldung



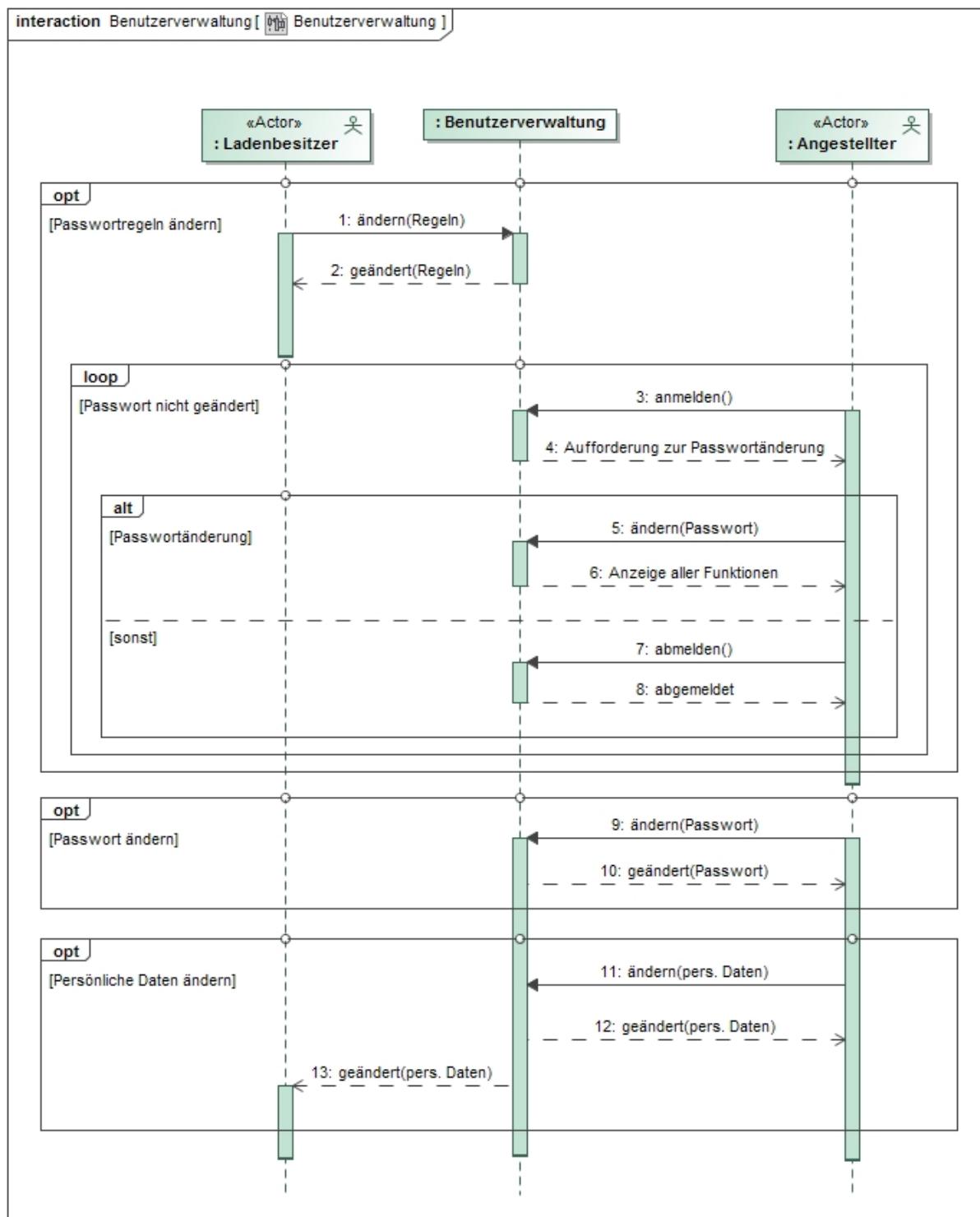
Dieses Diagramm beschreibt den Anmeldevorgang des Benutzers und wie das Programm bei einem fehlerhaften Benutzernamen oder Passwort reagiert.

Witze



Im oben stehenden Diagramm wird gezeigt, wie der Ladenbesitzer einen neuen Witz zur bestehenden Witzsammlung hinzufügen kann, wie ein zufälliger Witz aus der Sammlung einem Angestellten angezeigt wird und wie ein Witz aus der Sammlung gelöscht wird.

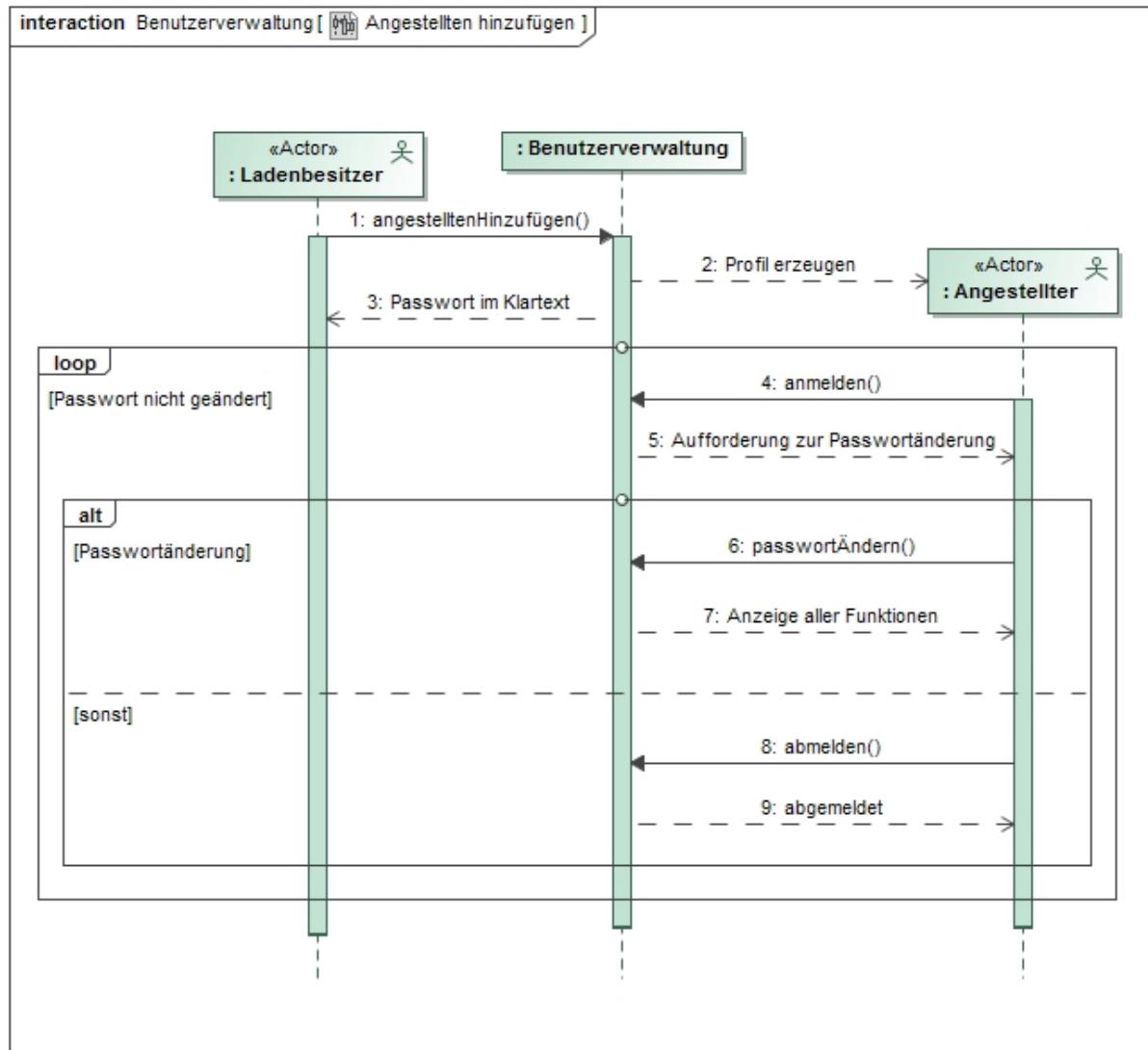
Benutzerverwaltung



In dem Sequenzdiagramm der Benutzerverwaltung werden die Möglichkeit der Änderung der persönlichen Daten und das Setzen neuer Passwortregeln sowie die dazugehörige Än-

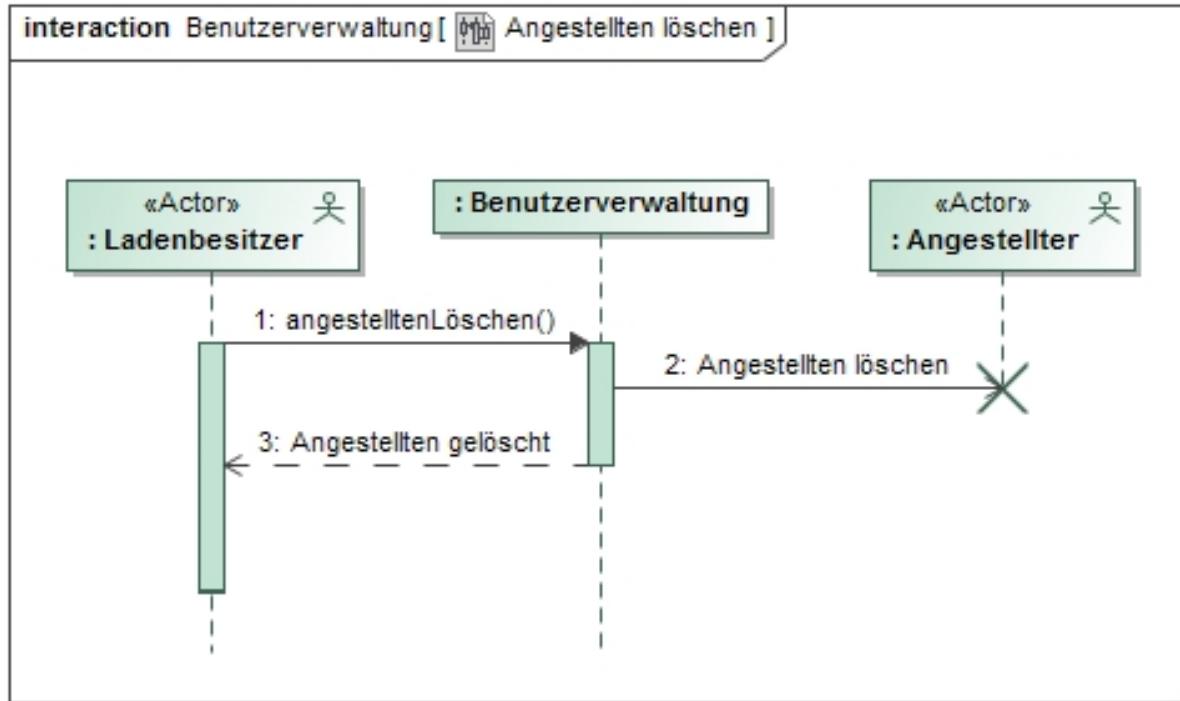
derung des Passwortes beschrieben.

Angestellten hinzufügen



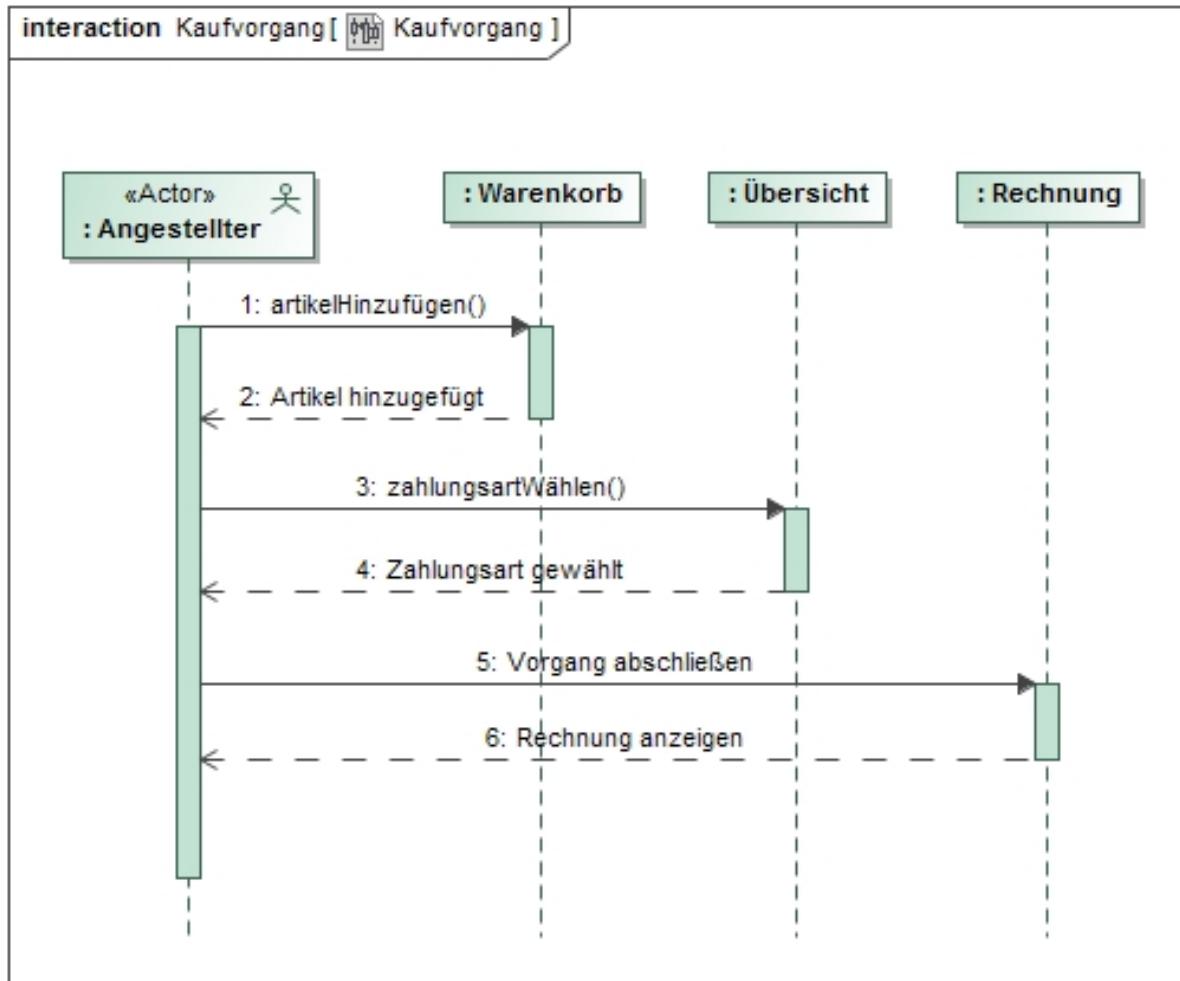
Das Sequenzdiagramm zeigt den Ablauf und die Interaktion zwischen den Objekten des Systems für den Fall, dass ein neuer Mitarbeiter angelegt wird.

Angestellten löschen



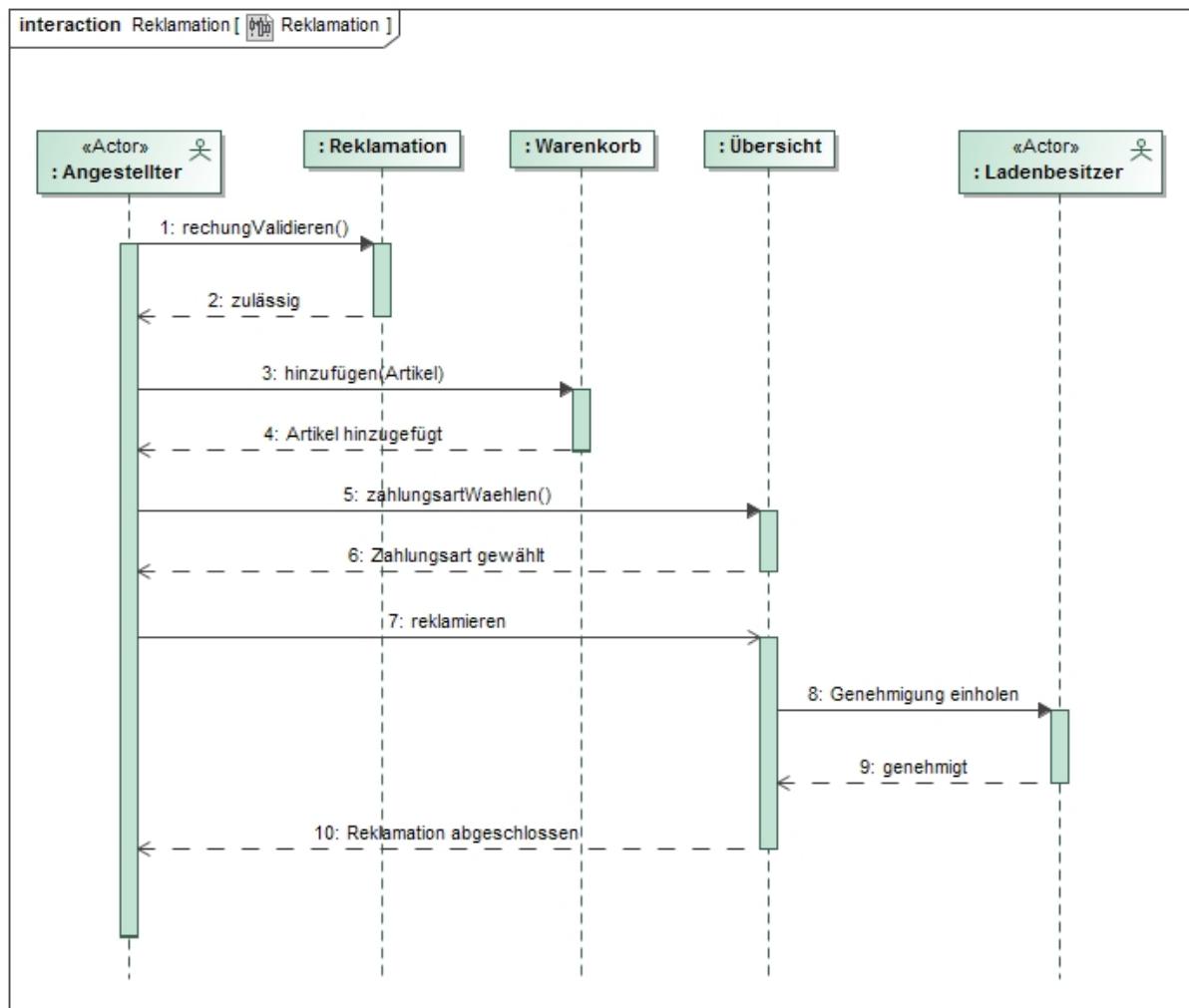
Das Sequenzdiagramm zeigt den Ablauf und die Interaktion zwischen den Objekten des Systems für den Fall, dass ein neuer Mitarbeiter gelöscht wird.

Kaufvorgang



Dieses Diagramm zeigt, wie ein üblicher Kaufvorgang abläuft.

Reklamation

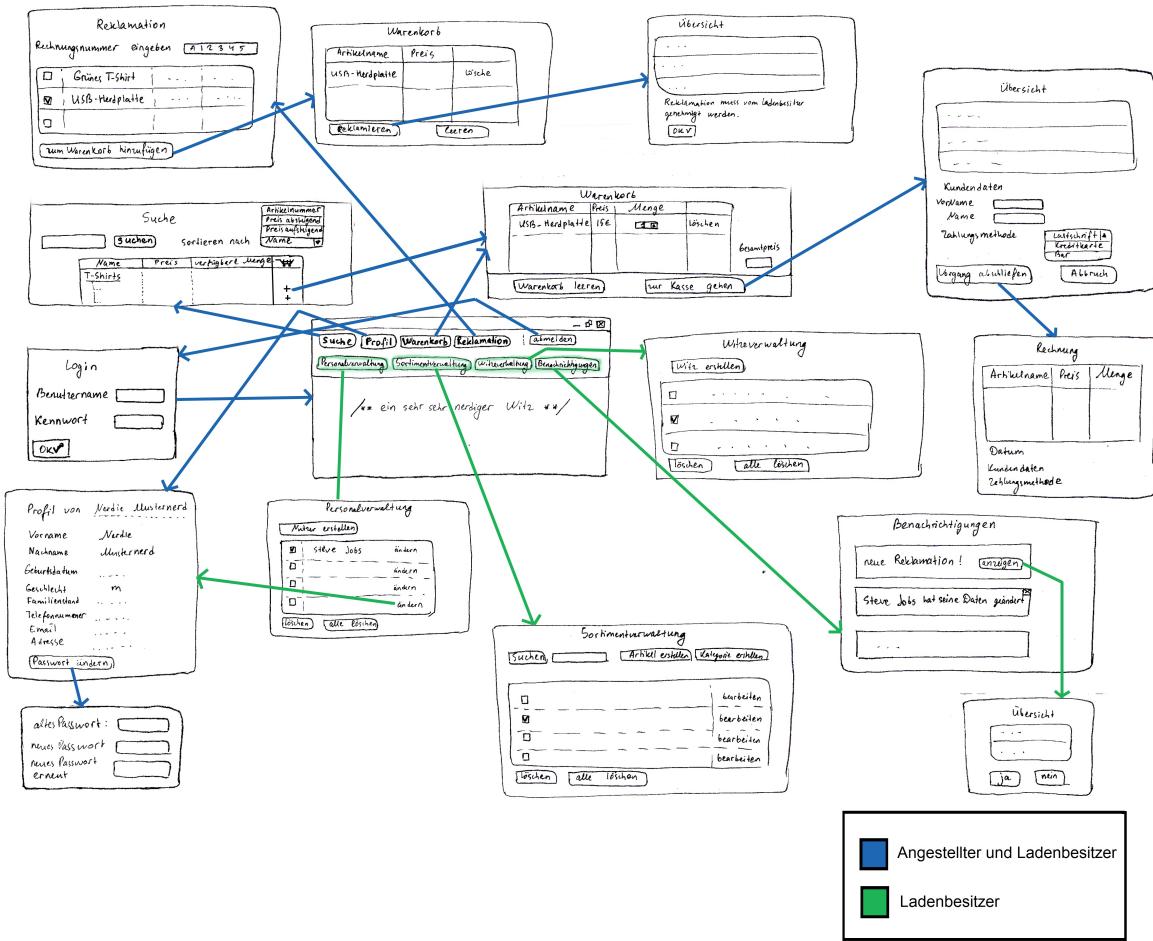


Hier wird beschrieben, wie eine Reklamation abläuft.

5.1 Persistenz

Die Persistenz wird mit Hilfe der Repositorys durch Spring und der in Maven integrierten Datenbank geregelt.

5.2 Benutzeroberfläche



5.3 Ergonomie

Die Benutzeroberfläche ist intuitiv gestaltet und leicht verständlich. Das Produkt wurde von der Partnergruppe mit gleichem Projekt im Crosstesting geprüft.

5.4 Transaktionsbehandlung

Es wurden die Transaktionen Barzahlung, Kreditkarte und Lastschrift implementiert.

5.5 Sessionbehandlung und Sicherheit

Anhand von einer Session werden eingelogte Nutzer unterschieden. Das System basiert auf Spring Security.

5.6 Plausibilisierung und Validierung

Alle Nutzereingaben werden überprüft.

5.7 Konfigurierbarkeit

Der Ladenbesitzer kann jederzeit die Passwortregeln ändern.

5.8 Internationalisierung

Es wird nur die Sprache Deutsch unterstützt.

5.9 Buildmanagement

Der vollständige Sourcecode ist auf GitHub vorhanden.