

# Software-Praktikum

SS 2021, Blatt T1



Tutorium: 30.08.2021

Prof. Dr. Sven Apel

**Hinweis:** Der Leitfaden für die von uns erwartete Notation bezüglich der UML-Diagramme wird in der Lösung zu Übungsblatt 1 zu finden sein.

## Aufgabe 1

Der Anforderungskatalog für den neuen Uni-Müsli Onlineshop lautet wie folgt:

- In einem Kundenkonto wird bei der Registrierung folgendes hinterlegt: Mailadresse, Passwort, Telefonnummer und Lieferadresse bestehend aus Vor- und Nachname, Straße, Hausnummer, PLZ und Ort.
- Man kann ohne Login anfangen, Müslis zusammenzustellen. Dafür wählt man eine Basis und danach die Toppings aus. Wenn das Müsli fertig ist, kann man es dem Warenkorb hinzufügen oder abrechnen.
- *Edit:* Erst nach erfolgreichem Login kann man seine persönlichen Daten ändern, Bestellhistorie ansehen und eine Bestellung aufgeben. Um die Bestellung abzuschließen, muss der Online-Bezahlvorgang erfolgreich abgeschlossen sein.
- Als *IngredientType* gibt es bisher *OATS*, *RAISIN*, *BERRIES*, *BANANA*, *CHOCOLATE* und *NUTS*.
- Der Onlineshop-Admin kann den Produktkatalog um Zutaten erweitern oder bestehende Zutaten bearbeiten/entfernen.

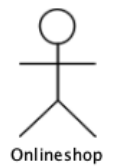
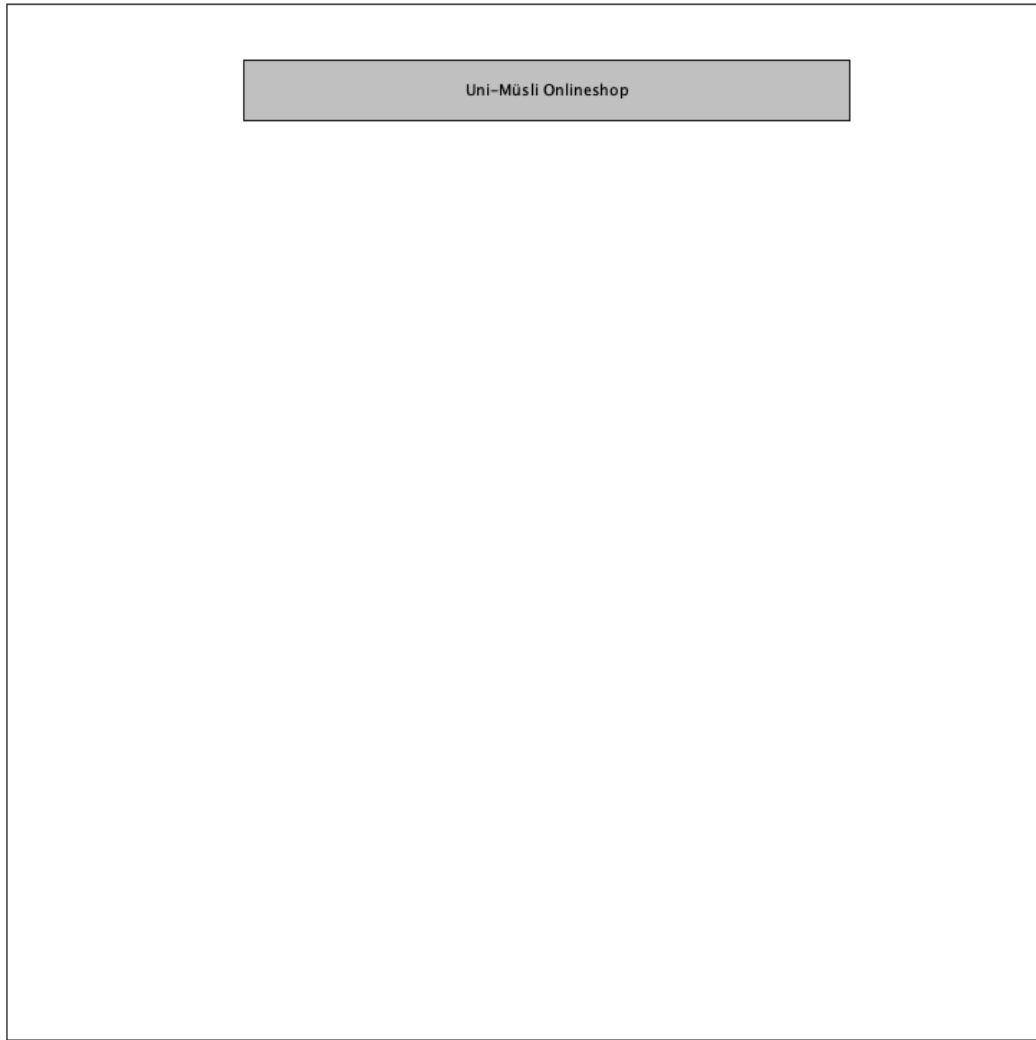
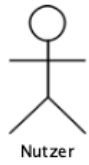
a) Ergänzen Sie das Use-Case Diagramm.

b) Für das UML-Klassendiagramm ergänzen Sie bitte sinnvoll Klassen, Felder und Methoden, um die Zusammenstellung und das Bestellen von Müsli abzubilden. Zeichnen Sie außerdem die Assoziationen zwischen den einzelnen Klassen ein.

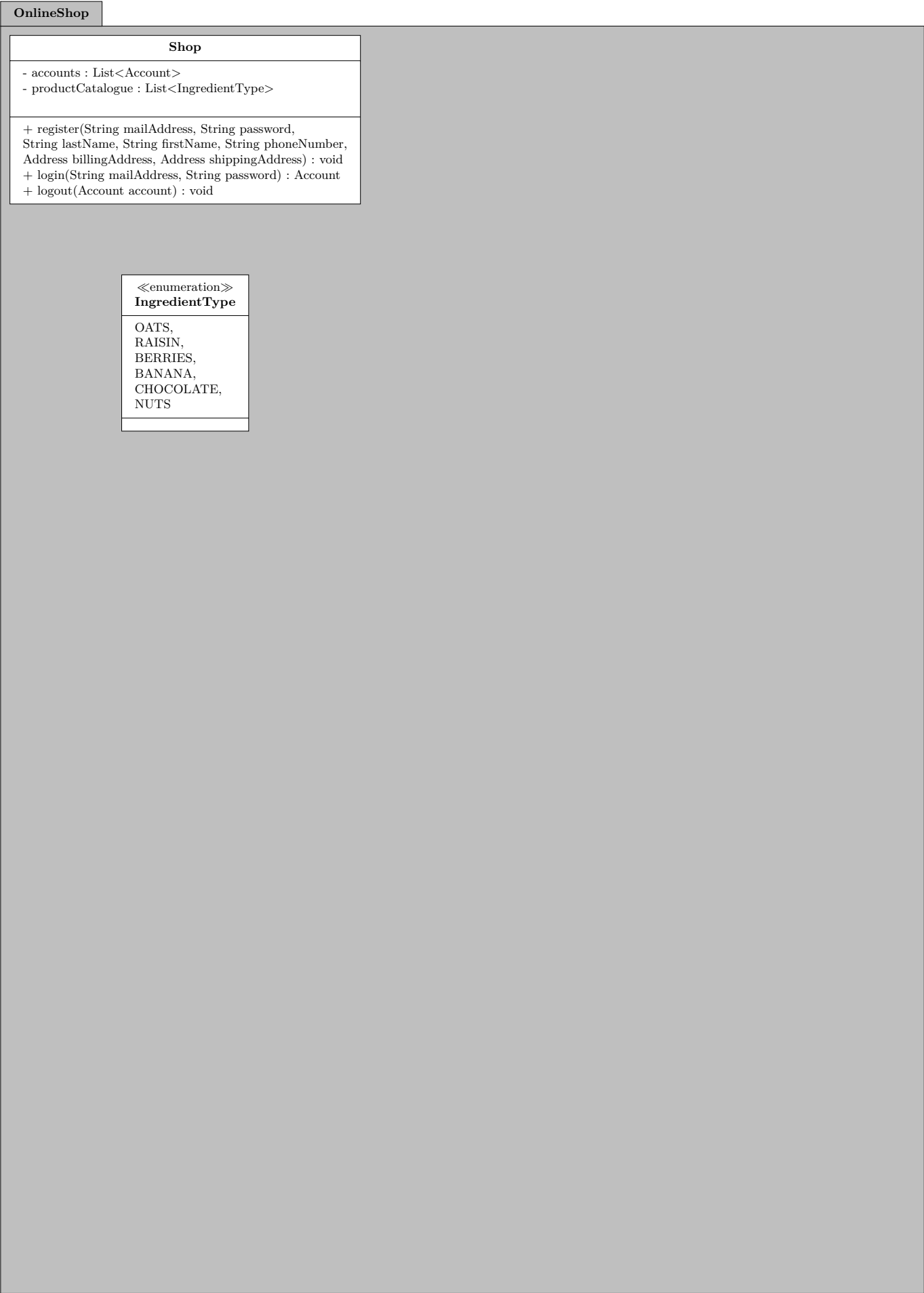
c) Erstellen Sie darauf aufbauend das Sequenzdiagramm für das Szenario *eine Bestellung aufgeben* :

Ein (eingeloggter) Nutzer stellt sich ein Müsli zusammen. Er wählt als Basis Haferflocken (*OATS*), als Toppings *BANANA* und *CHOCOLATE*. Danach fügt er das Müsli zum Warenkorb hinzu, geht zur Kasse, bezahlt und erhält eine Bestellbestätigung.

**Zu a):**

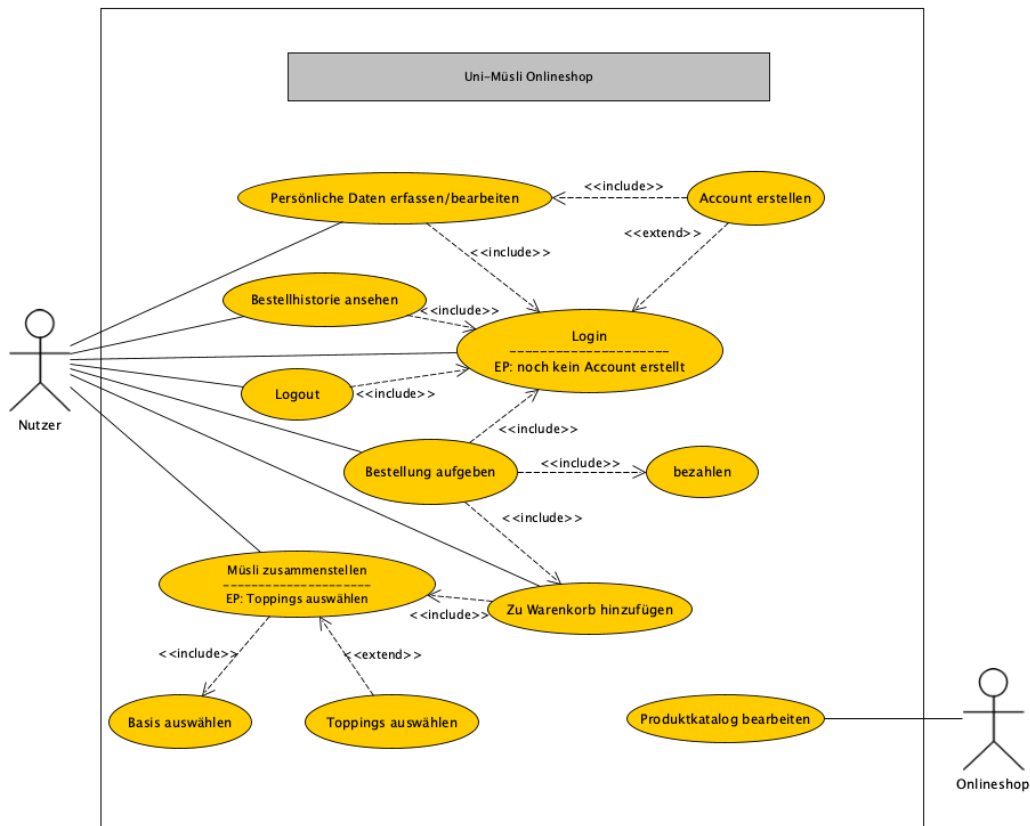


Zu b):



# Lösung

a)



*Hinweis:* Die Use-Case Diagramme von den Übungs- und Tutorienblättern decken alle für die Vorlesung relevanten Fälle und Beziehungen zwischen Use Cases hinreichend ab. Zur besseren Lesbarkeit sind die Assoziationen im Use Case als einfache Striche dargestellt, im Gegensatz zu der ebenfalls gängigen Variante des Strichs mit Pfeilen an beiden Enden.

## Allgemeine Vorgehensweise und Beschreibung des Lösungsvorschlags:

- Extrahiere Anwendungsfälle aus der Aufgabenbeschreibung (Bsp. „Müsli zusammenstellen“, „Basis auswählen“, „Produktkatalog bearbeiten“, usw.)
- Welche Anwendungsfälle sind miteinander logisch verknüpft und/oder voneinander abhängig?

### Definition: *include-Beziehung*

Eine include-Beziehung zwischen zwei Use Cases bedeutet, dass ein Use-Case für die vollständige und korrekte Ausführung das andere Use-Case benötigt.

Beispiel: „Bestellung aufgeben“ benötigt „Login“ für die korrekte Ausführung.

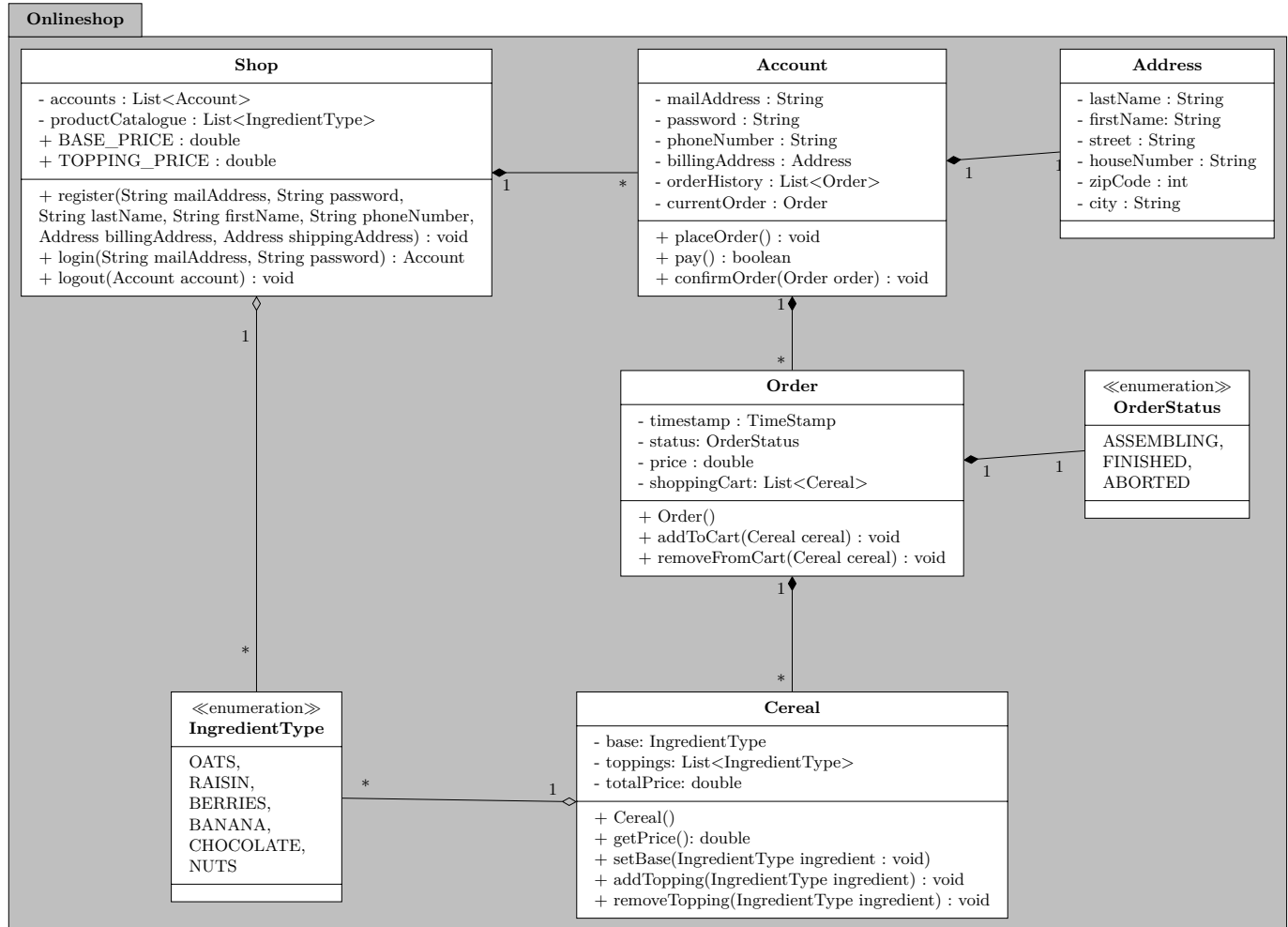
### Definition: *extend-Beziehung*

Eine extend-Beziehung zwischen zwei Use Cases bedeutet, dass ein Use-Case in einem Spezialfall (z.B. ein Fehler) die Ausführung eines anderen Use Cases auslöst. Der sog. *Extension Point (EP)* spezifiziert dabei kurz, in welchem Fall der extend Use-Case ausgelöst wird.

Beispiel: „Zu Warenkorb hinzufügen“ ist erst möglich, wenn das Müsli fertig zusammengestellt ist (= *Extension Point*).

- Was kann welcher Akteur ausführen? Wie kann welcher Akteur interagieren (= Verbindungen zwischen Use Cases und Akteur einzeichnen).

b)



c)

