

<b>Max-Eyth-Schule</b>	<b>Use-Case-Diagramm</b>	Klasse:
Tag:		Name:

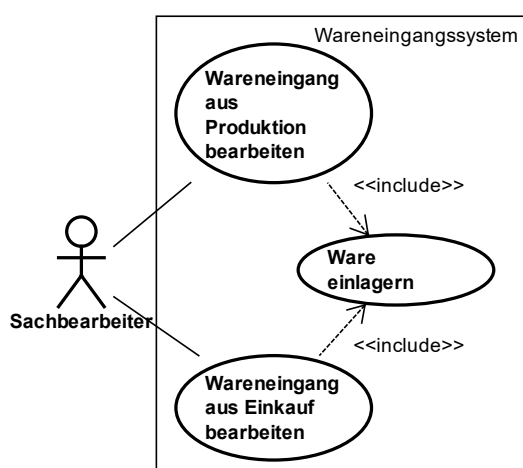
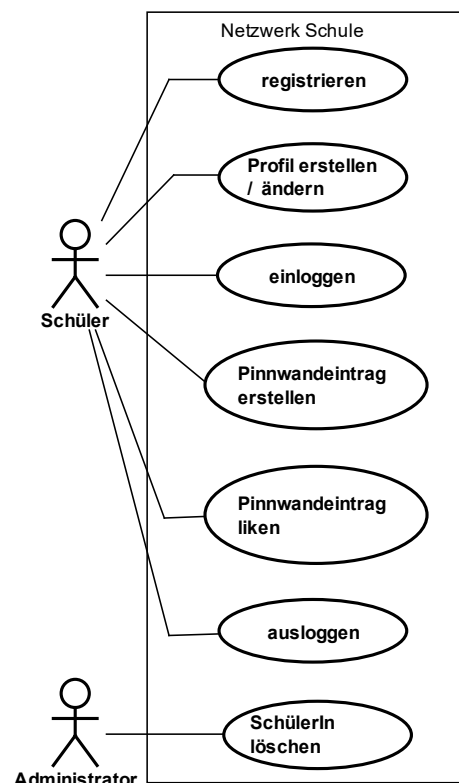
## Dokumentation der Anforderungen - das Use-Case-Diagramm

Use-Case-Diagramme werden zu Beginn des Softwareentwicklungsprozesses erstellt, um die Anforderungen an das System bildlich darzustellen. Es ist vorteilhaft Anforderungen möglichst frühzeitig zu dokumentieren. Fachliche Anforderungen, die beispielsweise erst während der Implementierung festgestellt werden, führen häufig zu kostenintensiven Änderungen des Systems. Grundsätzlich gilt, je später Anforderungen in einem Entwicklungsprozess bekannt werden, umso aufwändiger und teurer wird die Entwicklung.

Ein Use-Case-Diagramm beschreibt im Rahmen der Anforderungsanalyse "Was" ein System leisten soll. Die technische Implementierung, das "Wie" spielt dabei keine Rolle. Ein Use-Case-Diagramm stellt durch Anwendungsfälle (**use cases**) und **Akteure (actors)** die Funktionalität eines Softwaresystems dar. Es dient dazu, mit den zukünftigen Benutzern über diese Funktionalität zu sprechen, ohne sich gleich in Details zu verlieren. Ein Use-Case-Diagramm zeigt das zu entwickelnde System und das Zusammenspiel mehrerer Use-cases untereinander und mit den Akteuren, es gibt auf hohem Abstraktionsniveau einen guten Überblick über das System und seine Schnittstellen zur Umgebung<sup>1</sup>.

Ein **Use-case** (use case, Anwendungsfall) wird als Ellipse dargestellt und mit der Bezeichnung des Use-case beschriftet, diese Bezeichnung sollte ein Verb enthalten. Ein Use-case spezifiziert eine Sequenz von Aktionen, einschließlich möglicher Varianten, die das System in Interaktion mit Akteuren ausführt. Der Use-case wird durch ein bestimmtes Ereignis ausgelöst und ausgeführt um ein Ziel zu erreichen oder ein gewünschtes Ergebnis zu erzielen. Er beschreibt das extern wahrnehmbare Verhalten, ohne auf die interne Struktur oder Details der Realisierung einzugehen.

Ein **Akteur** (actor) ist eine Rolle, die ein Benutzer des Systems einnimmt. Jeder Akteur hat einen gewissen Einfluss auf das System. Ein Akteur ist häufig eine Person und wird dann als Strichmännchen dargestellt und mit der Rolle beschriftet. Es kann sich ebenso um eine Organisationseinheit oder ein externes System handeln, das mit dem zu modellierenden System kommuniziert. In diesem Fall wird er als Rechteck dargestellt. Die Kommunikation wird in Form einer durchgezogenen Linie vom Akteur zum Use-case dargestellt. Die Darstellung der Use-cases stellt keine Reihenfolge der Abarbeitung dar! Zwischen Use-cases können mehrere Arten von Beziehungen existieren. Die **include**-Beziehung macht es möglich, gemeinsame Funktionalität aus zwei Use-cases in einen Dritten auszulagern, den die beiden ersten Use-Cases dann inkludieren. Der dritte Use-case wird immer für die korrekte Ausführung der ersten beiden benötigt. Wenn der erste Use-case ausgeführt wird, wird der dritte auch immer ausgeführt, gleiches gilt für den zweiten. Die include-Beziehung erspart die mehrmalige Beschreibung desselben Verhaltens. Sie wird durch einen gestrichelten Pfeil mit offener Spitze dargestellt, an der Linie steht <<include>>, die Pfeilspitze zeigt zum dritten, ausgelagerten Use-case.



<sup>1</sup> Balzert, Heide, Lehrbuch der Objektmodellierung, S. 62

<b>Max-Eyth-Schule</b>	<b>Use-Case-Diagramm</b>	Klasse:
Tag:		Name:

Mit Hilfe der **extend**-Beziehung wird ein Use-case durch einen anderen Use-case erweitert. Der erste Use-case beschreibt die Basisfunktionalität, der andere spezifiziert eine Erweiterung. So ist es möglich, bei komplexen Use-cases zuerst die Basisfunktionalität zu spezifizieren und ihn dann nach und nach zu erweitern. Der Basis- Use-case kann allein oder zusammen mit den Erweiterungen ausgeführt werden. **Extension points** geben an, an welchen Stellen der Basis-Use-case erweitert wird. Erweiterungen sind an **Bedingungen** geknüpft, diese Bedingung wird zusammen mit dem zugehörigen extension point auf einen Notizzettel geschrieben und an die extend-Beziehung geheftet.

Die extend-Beziehung wird durch einen gestrichelten Pfeil mit offener Spitze dargestellt, an der Linie steht `<<extend>>`, die Pfeilspitze zeigt zu dem Use-case, der erweitert wird, nicht zu Erweiterung. Bedingung und die Bezeichnung des extension points stehen auf dem Notizzettel.

**Generalisierung von Akteuren:** Zwischen einem allgemeinen Akteur und einem spezialisierten Akteur kann eine Generalisierung definiert werden, die durch einen durchgezogenen Pfeil mit Dreiecksspitze vom spezialisierten zum generalisierten Akteur dargestellt wird. Der spezialisierte Akteur erbt alle Kommunikationsbeziehungen des generalisierten, ohne dass diese explizit ins Diagramm eingezeichnet werden. Im Beispiel der Auftragsbearbeitung bedeutet das, dass Sachbearbeiter und Abteilungsleiter einen Auftrag ausführen und Artikel beim Lieferant bestellen können, aber nur der Abteilungsleiter bearbeitet Reklamationen.

In der Literatur zu UML findet man auch die Generalisierung von Use-cases zur Darstellung hierarchischer Zusammenhänge, das ist für unsere Zwecke bisher nicht relevant.

Hier ein umfangreicheres Beispiel eines Systems zur Essensbestellung in einer Kantine:

