

# ÜBUNG SZETTEL UML

## OBJEKTORIENTIERTE ANALYSE

Arbeite dich in die Grundprinzipien der Objektorientierten Analyse (OOA) ein.

- Ziele
- Requirements-Engineering
- Strukturelle Modellierung
- Dynamische Modellierung

[https://de.wikipedia.org/wiki/Objektorientierte\\_Analyse\\_und\\_Design#Objektorientierte\\_Analyse](https://de.wikipedia.org/wiki/Objektorientierte_Analyse_und_Design#Objektorientierte_Analyse)

Dokumentiere die wesentlichen Aspekte sowie deine persönlichen Lernergebnisse.

Nach der Beschäftigung mit dieser Aufgabe musst du wissen, was OOA ist, wozu man sie braucht, wie und wann man sie im Softwareentwicklungsprozess einsetzt.

## ANFORDERUNGSANALYSE MIT ANWENDUNGSFÄLLEN

Arbeite dich in die Grundprinzipien der Anforderungsanalyse mit USE-Cases (Diagramm + textuelle Beschreibung + Mockup) ein.

- Grundlagen
- Vorgehensweisen
- Werkzeuge
- Dokumentation

Dokumentiere die wesentlichen Aspekte sowie deine persönlichen Lernergebnisse.

Nach der Beschäftigung mit dieser Aufgabe musst du wissen, was unter Anforderungsanalyse zu verstehen ist, welche Werkzeuge man dazu verwendet und wie man diese einsetzt. Insb. musst du wissen, wie man USE-Case-Diagramme zeichnet, wie man Use-Cases genau beschreibt und welche Rolle UI-Mockups in diesem Zusammenhang spielen.

Ressourcen zum Start:

- [https://de.wikipedia.org/wiki/Objektorientierte\\_Analyse\\_und\\_Design#Objektorientierte\\_Analyse](https://de.wikipedia.org/wiki/Objektorientierte_Analyse_und_Design#Objektorientierte_Analyse)

## ANFORDERUNGSANALYSE MIT USER STORIES UND EPICS

Arbeite dich in die Grundprinzipien der Anforderungsanalyse mit User-Stories und Epics ein.

- Grundlagen
- Vorgehensweisen
- Werkzeuge
- Dokumentation

Dokumentiere die wesentlichen Aspekte sowie deine persönlichen Lernergebnisse.

Ressourcen zum Start:

- <https://www.atlassian.com/agile/project-management/epics>

- <https://www.atlassian.com/agile/project-management/user-stories>
- <https://www.atlassian.com/agile/project-management/epics-stories-themes>
- <https://miro.com/miroverse/user-stories-recipes/>
- <https://www.youtube.com/watch?v=ogV2r9579WI>

## WICHTIGE UML-DIAGRAMMARTEN

Arbeite dich in folgende Diagrammarten ein:

- Klassendiagramm
- Sequenzdiagramm
- Zustandsdiagramm

Fasse für jede Diagrammart folgende Punkte übersichtlich zusammen:

- Einsatzzweck
- Wesentliche Notationsformen
- Exemplarische Anwendung anhand eines Beispiels

Nach der Beschäftigung mit dieser Aufgabe musst du wissen, wie man die gegebenen Diagrammarten korrekt zeichnet und wann man sie braucht.

## TEXTUELLES DIAGRAMM-DESIGN-TOOL

Arbeiten Sie sich in die beiden folgenden Diagramm-Design-Tools ein. Du verwendest sie für die folgenden Aufgaben.

PlantUML

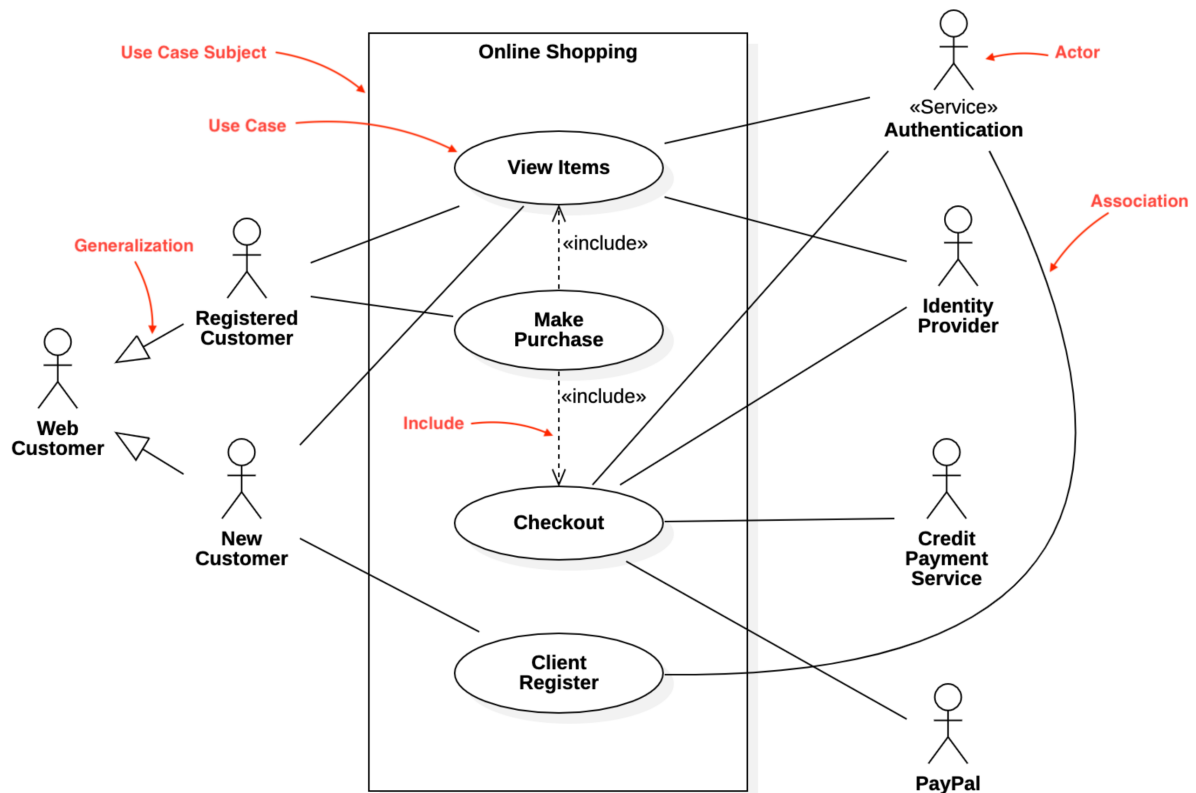
- <https://plantuml.com/de/use-case-diagram>

Mermaid

- <https://mermaid.js.org/intro/>
- <https://mermaid.live/>

## USE-CASE-DIAGRAMM 1

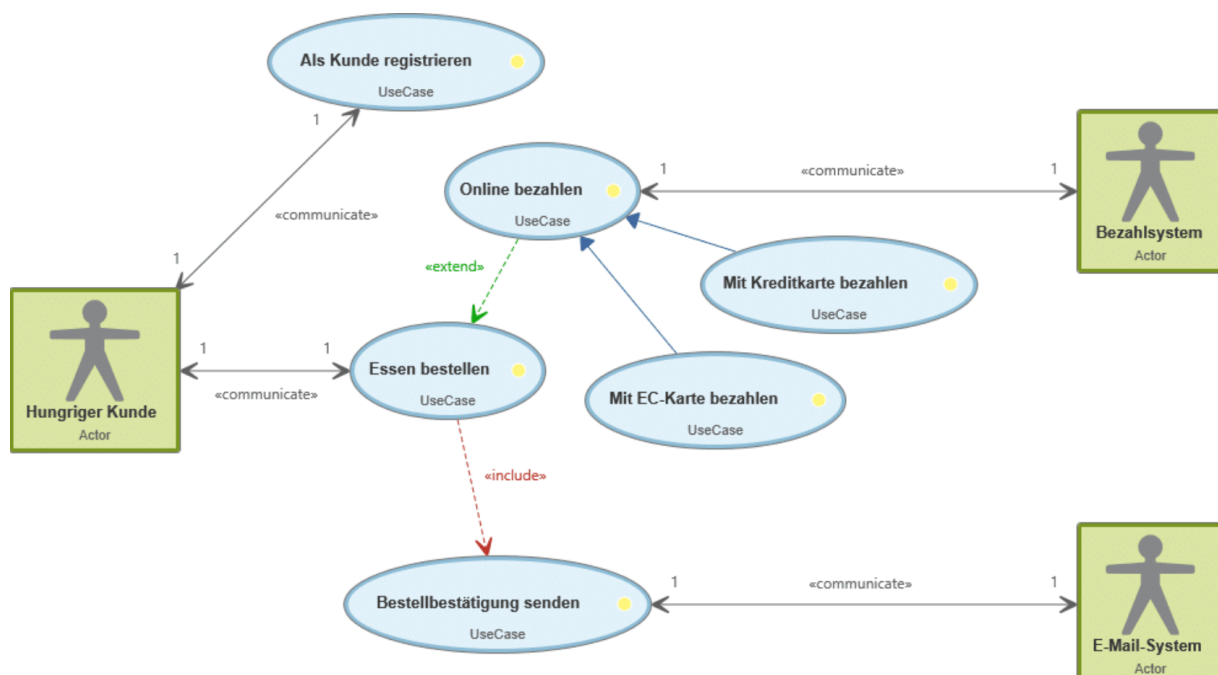
Interpretieren Sie schriftlich das folgende Use-Case-Diagramm:



<https://docs.staruml.io/working-with-uml-diagrams/use-case-diagram>

## USE-CASE-DIAGRAMM 2

Interpretieren Sie schriftlich das folgende Use-Case-Diagramm:



## USE-CASE-DIAGRAMM 3

Entwerfen Sie ein Anwendungsfalldiagramm zu der folgenden Beschreibung:

In einem Kino kann ein Gast Kinokarten an der Kasse kaufen, die vorbestellt sein könnten. Außerdem ist es möglich Popcorn und Getränke zu bestellen. Danach bezahlt der Kunde beim Kassierer die Rechnung. Es ist auch möglich mit Kreditkarte zu bezahlen, welche bei Bedarf einer automatischen Prüfung unterzogen werden kann.

## USE-CASE-DIAGRAMM 3

Entwerfen Sie ein Anwendungsfalldiagramm zu der folgenden Beschreibung:

Es soll ein Anwendungssystem zur Unterstützung der Geschäftsprozesse in einem Krankenhaus entwickelt werden. Das System soll folgende Aufgaben erledigen:

- Herr Müller und Herr Maier seien in der Verwaltung angestellt.
- Herr Müller soll Mitarbeiter einstellen und entlassen können. Sowohl Herr Müller als auch Herr Maier kann Patienten aufnehmen und entlassen.
- Sowohl bei der Einstellung von Mitarbeitern, als auch bei der Aufnahme von Patienten müssen Name und Adresse erfasst werden. Um redundante Anwendungsfall-Beschreibungen zu verhindern wird diese Tätigkeit in einen gesonderten Anwendungsfall ausgelagert.
- Falls der einzustellende Mitarbeiter bzw. der aufzunehmende Patient seinen Wohnsitz am Ort des Krankenhauses hat, wird geprüft, ob die angegebene Adresse am Wohnort existiert. Lagern Sie auch diesen Anwendungsfall aus.

## USE-CASE-DETAILBESCHREIBUNGEN

Definieren Sie die Use-Case-Details (level, complexity, status, pre-conditions, post-conditions and assumptions, event flow etc.) für einige Use-Cases der vorhergehenden Use-Case-Übungen. Verwenden Sie dazu eine der bereitgestellten Schablonen bzw. ein entsprechendes Software-Tool.

## Schablone A

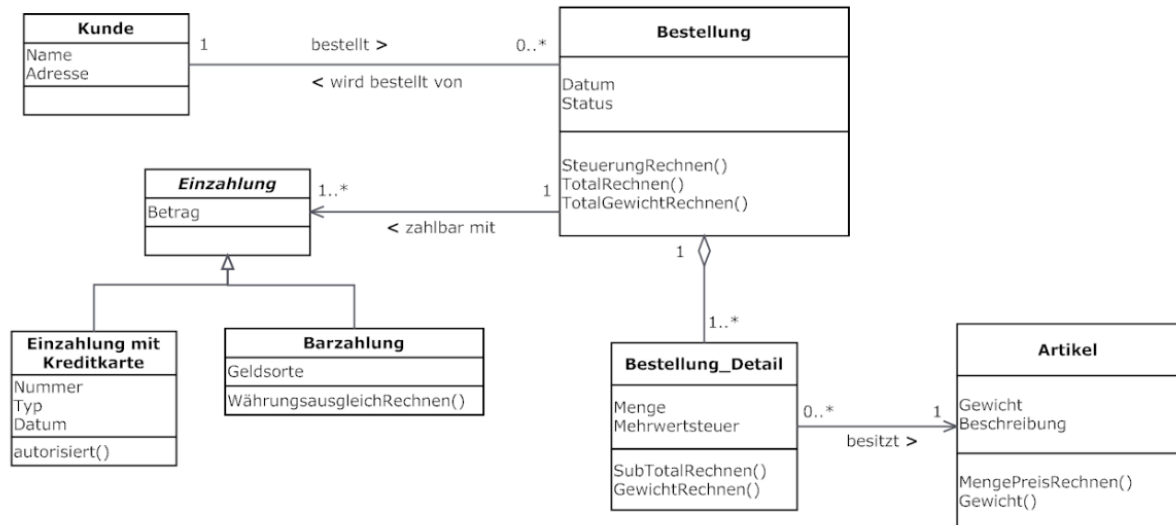
<b>Name</b>	Taxi bestellen
<b>Ziel im Kontext</b>	Ein Kunde bestellt ein Taxi
<b>Akteure</b>	Kunde, Taxivermittler
<b>Trigger</b>	Kunde ruft den Taxivermittler an
<b>Essenzielle Schritte</b>	<ol style="list-style-type: none"> <li>1. Ein Kunde ruft den Taxivermittler an, um ein Taxi zu bestellen.</li> <li>2. Der Kunde teilt dem Taxivermittler die Bestelldetails mit.</li> <li>3. Der Taxivermittler registriert den Auftrag im System.</li> <li>4. Der Taxivermittler bestätigt dem Kunden den Bestellauftrag.</li> </ol>
<b>Erweiterungen</b>	<ol style="list-style-type: none"> <li>3a. Der Taxivermittler stellt fest, dass kein Taxi verfügbar ist:               <ol style="list-style-type: none"> <li>3a1. Der Taxivermittler stellt eine Anfrage an den übergeordneten Taxiverband</li> <li>3a2. Der Taxivermittler erhält die Bestätigung vom Taxiverband.</li> </ol> </li> <li>3b. Der Taxivermittler stellt fest, dass der Kunde nicht vertrauenswürdig ist:               <ol style="list-style-type: none"> <li>3b1. Der Taxivermittler lehnt den Auftrag ab.</li> </ol> </li> </ol>

## Schablone B

<b>Name</b>	Geschwindigkeit regeln	
<b>Akteur</b>	Fahrer	
<b>Trigger</b>	Fahrer wählt Zielgeschwindigkeit aus	
<b>Kurzbeschreibung</b>	Der Tempomat regelt die durch den Fahrer eingestellte Geschwindigkeit durch Signale an die Motorsteuerung. Er kann dabei die Geschwindigkeit bis zu einer Zielgeschwindigkeit reduzieren oder erhöhen oder eine Wunschgeschwindigkeit halten.	
<b>Vorbedingungen</b>	Geschwindigkeit > 50 km/h, Tempomat eingeschaltet	
<b>Essenzielle Schritte</b>	<b>Intention der Systemumgebung</b>	<b>Reaktion des Systems</b>
	Fahrer startet Motor	Tempomat ist bereit
	Fahrer will Geschwindigkeit halten	Tempomat hält die Geschwindigkeit
	Fahrer will beschleunigen	Tempomat beschleunigt Fahrzeug
	Fahrer will Tempo reduzieren	Tempomat verringert Geschwindigkeit
	Timer löst Anzeige von Ziel- und Istgeschwindigkeit aus	Tempomat meldet Ziel- und Istgeschwindigkeit
	Fahrer übersteuert Tempomat (durch Kickdown, Vollbremsung, Geschwindigkeit < 50 km/h)	Regelung beenden
	Fahrer stellt Motor ab	Tempomat ausschalten
<b>Ausnahmefälle</b>	Defektbedingte Einschränkung der Funktionalität des Tempomaten.	
<b>Nachbedingung</b>	Tempomat ist ausgeschaltet.	
<b>Zeitverhalten</b>	Maximal 32 ms von der Eingabe der Zielgeschwindigkeit bis zur Aktualisierung der Anzeige inklusive der Ermittlung der benötigten Maßnahmen und Ansteuerung der betroffenen Nachbarsysteme.	
<b>Verfügbarkeit</b>	Maximal ein Systemausfall innerhalb 1000 Betriebsstunden.	
<b>Fragen, Kommentare</b>	Ist es möglich, die Werkseinstellungen des Tempomaten auf eigene Wünsche anzupassen (langsames Beschleunigen im Bereich über 120 km/h, Maximalgeschwindigkeit festlegen?)	

## KLASSENDIAGRAMME 1

Gegeben ist folgendes Klassendiagramm:



Bestimmen Sie, ob die folgenden Aussagen zum Klassendiagramm richtig oder falsch sind.

- ☐ Es kann im System Kunden geben die nie eine Bestellung durchgeführt haben.
- ☐ Die Klasse *Einzahlung* ist die Oberklasse der Klasse *Bestellung*.
- ☐ Jedes Objekt der Klasse *Bestellung\_Detail* besitzt genau einen Artikel.
- ☐ Alle *Einzahlungen mit Kreditkarte* haben einen Betrag.
- ☐ Es ist möglich, dass ein Artikel keine Assoziation mit einem *Bestellung\_Detail* besitzt.
- ☐ Jedes *Bestellung\_Detail*, das Teil einer Bestellung ist, hat seinen eigenen Status und sein eigenes Datum.

## KLASSENDIAGRAMME 2

Gegeben ist der folgende Sachverhalt.

Jede Person hat einen Namen, eine Telefonnummer und E-Mail. Jede Wohnadresse wird von nur einer Person bewohnt. Es kann aber sein, dass einige Wohnadressen nichtbewohnt sind. Den Wohnadressen sind je eine Strasse, eine Stadt, eine PLZ und ein Land zugeteilt. Alle Wohnadressen können bestätigt werden und als Beschriftung (für Postversand) gedruckt werden. Es gibt zwei Sorten von Personen: Student, welcher sich für ein Modul einschreiben kann und Professor, welcher einen Lohn hat. Der Student besitzt eine Matrikelnummer und eine Durchschnittsnote.

Modellieren Sie diesen Sachverhalt mit einem UML Klassendiagramm.

## KLASSENDIAGRAMME 3

Sie haben den Auftrag, eine Online-Videothek zu realisieren. Sie haben dazu folgende Angaben erhalten:

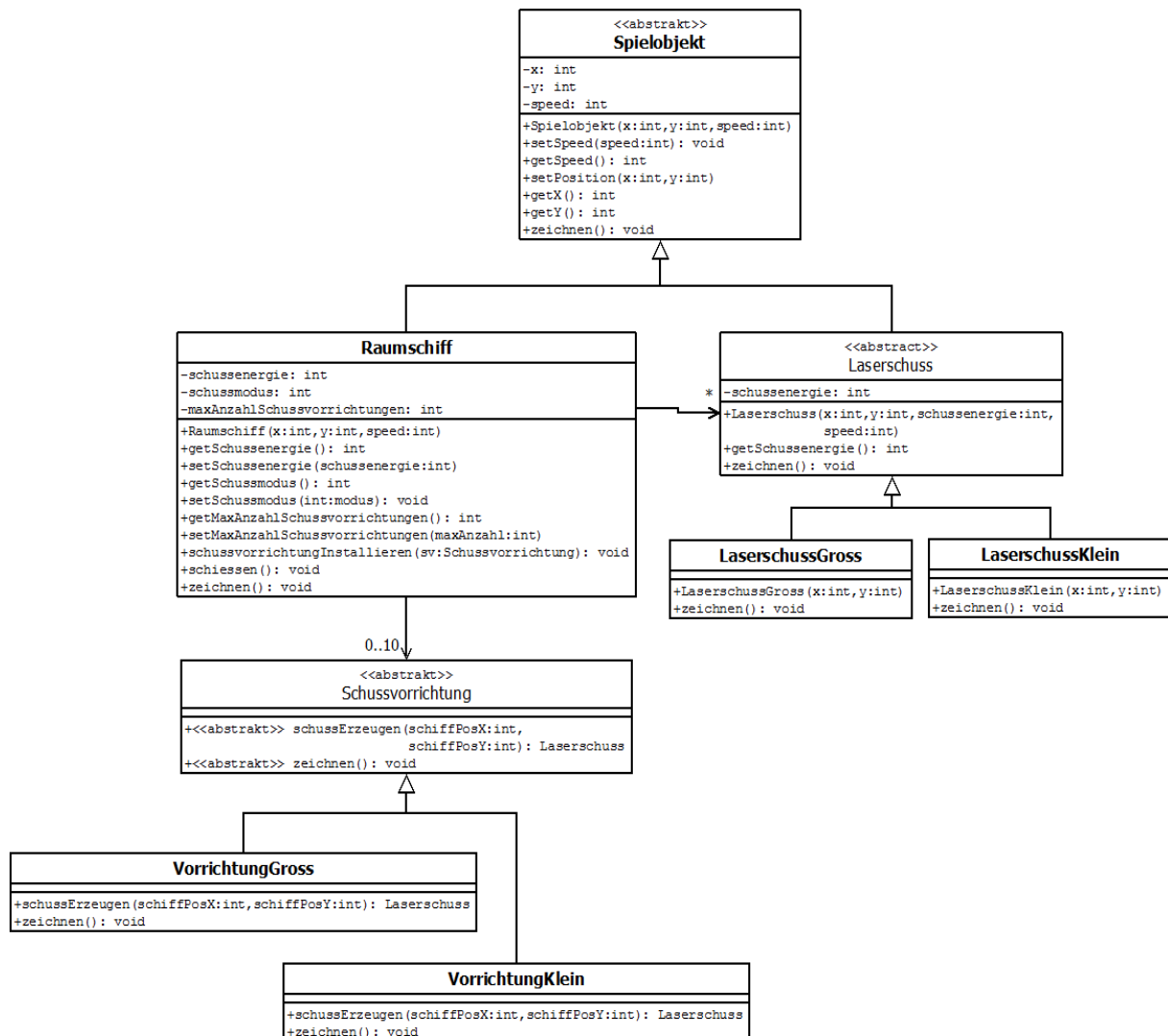
- Die Videothek unterstützt das Ausleihen von Filmen für registrierte Kunden. Dazu müssen
- Kunden sich zunächst mit ihrer Kundennummer und ihrem Passwort anmelden.
- Kunden werden zusammen mit ihrem Guthaben verwaltet.
- Filme besitzen einen individuellen Namen und Preis.

- Ein Film wird über einen Streaming-Server bereitgestellt. Der Server kann hierzu einen kundenspezifischen Link generieren.

Modellieren Sie diesen Sachverhalt anhand eines Klassendiagramms. Wählen Sie sinnvolle Operationen (mit möglichst vollständigen Signaturen) und Attribute für Ihre Klassen. Ergänzen Sie die Klassen um sinnvolle Beziehungen und deren Kardinalitäten.

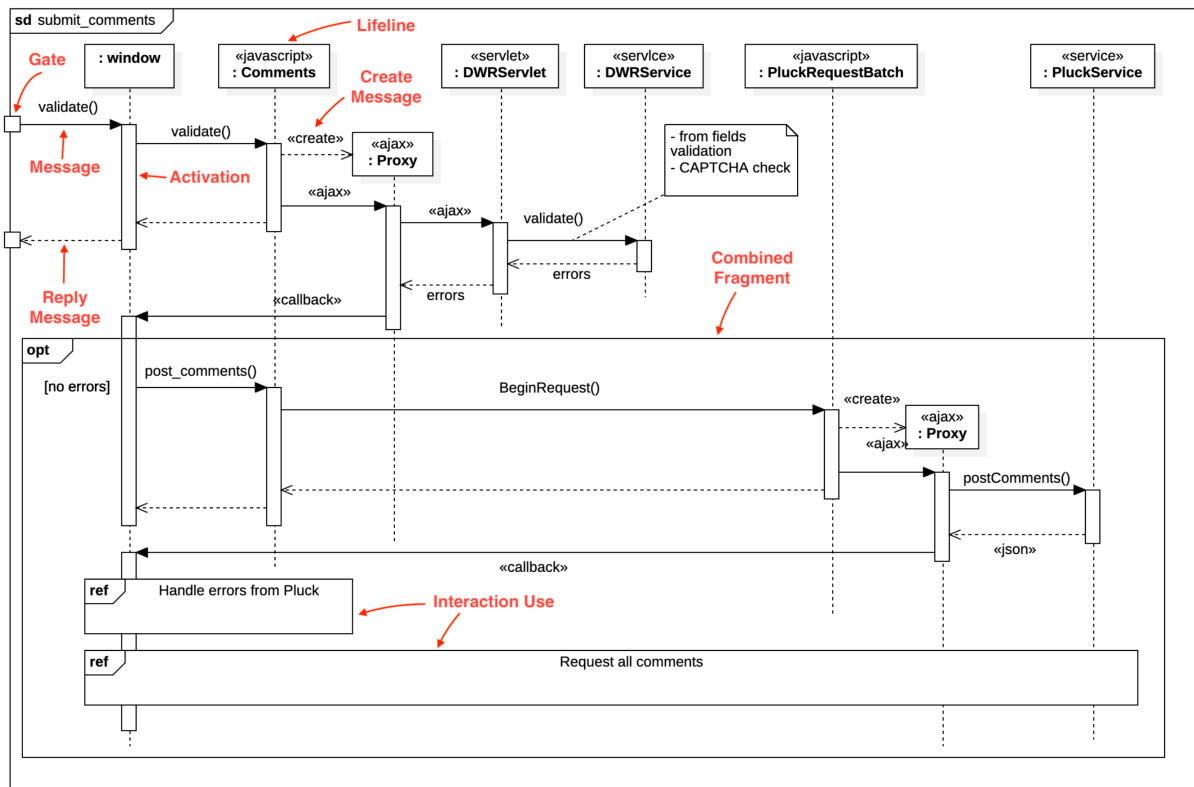
#### KLASSENDIAGRAMME 4

Interpretieren Sie schriftlich das folgende Klassendiagramm:



#### SEQUENZDIAGRAMM 1

Interpretieren Sie schriftlich das folgende Sequenzdiagramm:



<https://docs.staruml.io/working-with-uml-diagrams/sequence-diagram>

## SEQUENZDIAGRAMM 2

Modellieren Sie für die Online-Videothek (siehe Aufgabe 3) die Film Ausleihen Funktion. Erstellen Sie dazu ein Sequenzdiagramm für folgenden Ablauf der Ausleihe:

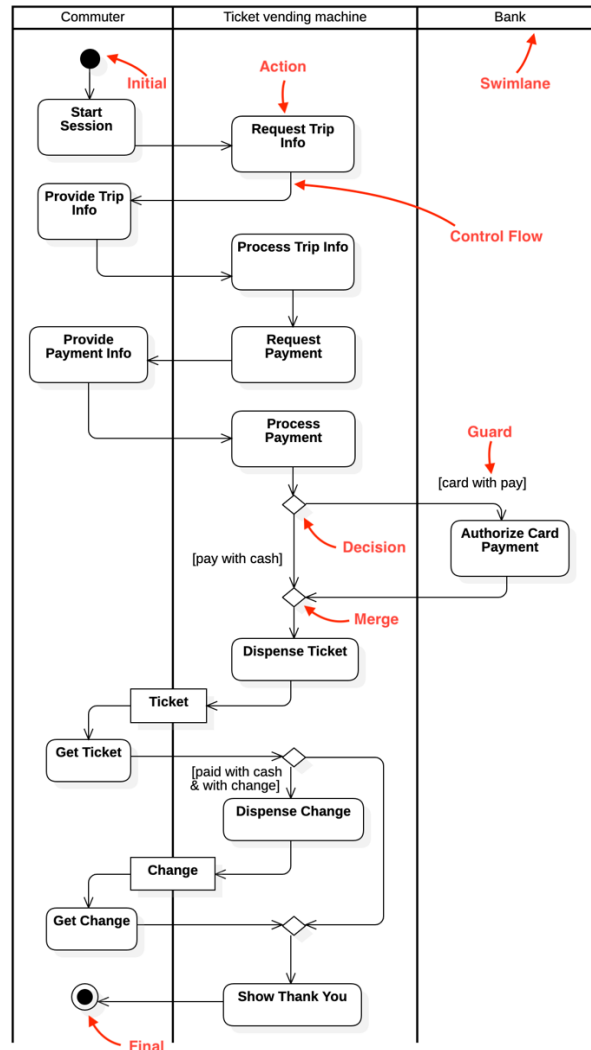
- Die Videothek berechnet zuerst, ob das Guthaben des Kunden reicht um den Film zu bezahlen.
- Reicht das Guthaben nicht aus, wird stattdessen eine Aufforderung zum Ausfüllen des Guthabens angezeigt.
- Falls das aktuelle Guthaben des Mitglieds ausreicht, veranlasst die Videothek einen Streaming-Server einen Link für den Film zu generieren.
- Die Videothek zeigt dem Benutzer den Link an, unter dem der Film zugreifbar ist.

Gehen Sie davon aus, dass sich das Mitglied bereits auf der Seite des gewünschten Films befindet.

## AKTIVITÄTSDIAGRAMM 1

Interpretieren Sie schriftlich das folgende Aktivitätsdiagramm:





<https://docs.staruml.io/working-with-uml-diagrams/activity-diagram>

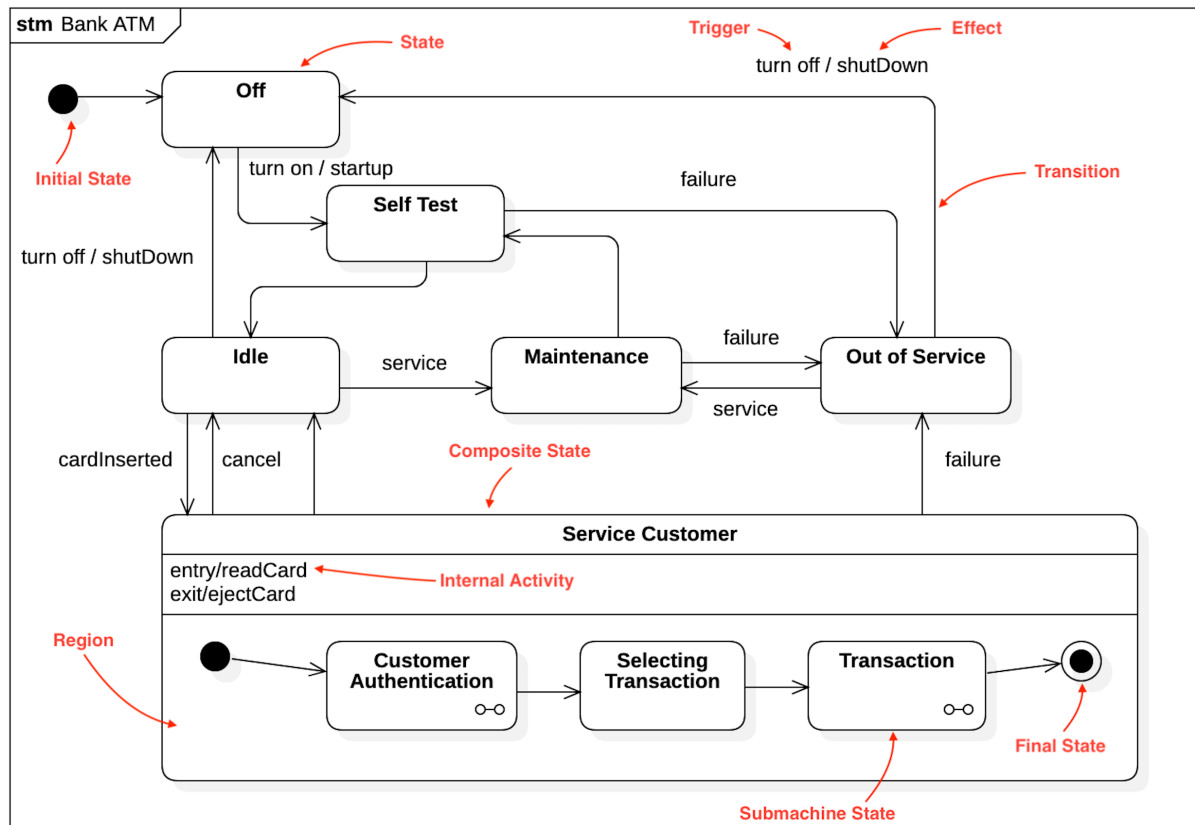
## AKTIVITÄTSDIAGRAMM 2

Modellieren Sie schriftlich den folgenden Sachverhalt als Aktivitätsdiagramm:

Ein Fluggast ist am Flughafen angekommen. Zur Überprüfung seines Tickets begibt er sich zum Schalter seiner Fluggesellschaft. Falls das Ticket in Ordnung ist, übergibt er am Schalter sein Gepäck. Falls mit dem Ticket etwas nicht stimmt, muss der Fluggast den Kundendienst konsultieren und er kann nicht mitfliegen. Das Gepäck wird zudem auf Übergewicht überprüft. Falls dem so ist, muss der Fluggast zusätzliche Kosten übernehmen. Falls aber das Gewicht in Ordnung ist, wird die Bordkarte ausgestellt.

## ZUSTANDSDIAGRAMM 1

Interpretieren Sie schriftlich das folgende Zustandsdiagramm:



<https://docs.staruml.io/working-with-uml-diagrams/statechart-diagram>

## ZUSTANDSDIAGRAMM 2

Entwerfen Sie ein Zustandsdiagramm für eine Bestellung auf Amazon. Modellieren Sie dazu die Zustände und die Übergänge einer Bestellung vom Aufgeben der Bestellung bis hin zur Aushändigung des Paketes an den Kunden.

## C4-DIAGRAMM

Arbeite dich in das C4-Modell zur Visualisierung von Architekturen von Softwaresystemen ein.

<https://c4model.com>

Dokumentieren Sie danach die Architektur deines Spring-Boot-Abschlussprojektes aus dem Unterricht mit dem C4-Diagrammmodell:

- Level 1: System Context diagram
- Level 2: Container diagramm
- Level 3: Component diagramm
- Level 4: Klassendiagramm (Beispielhaft nur für ein paar Komponenten aus Level 3)