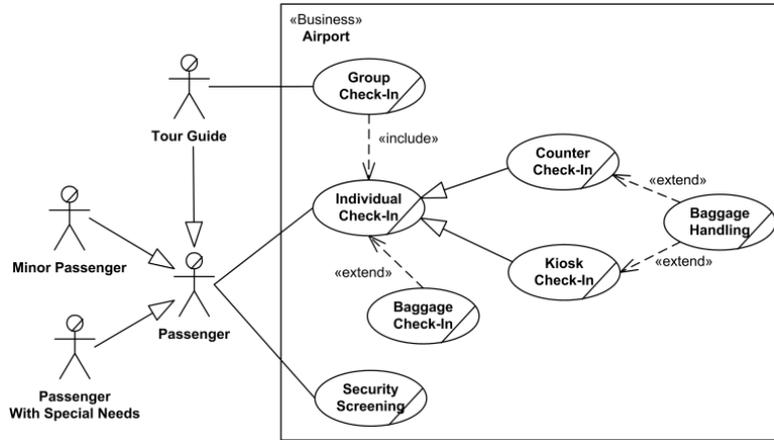


UML - Anwendungsfalldiagramme

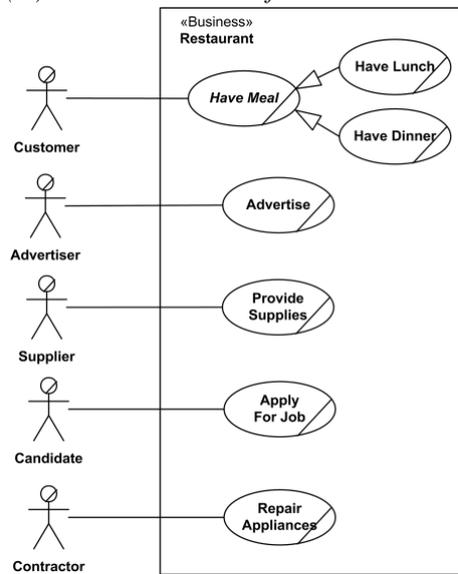
(1.) Geben Sie den Zweck von Anwendungsfalldiagrammen an!

(2.) Beschreiben Sie an Hand der in (a.)-(h.) wiedergegebenen Anwendungsfalldiagramme die wesentlichen Notationselemente und den Unterschied zwischen extend- und include-Beziehungen!

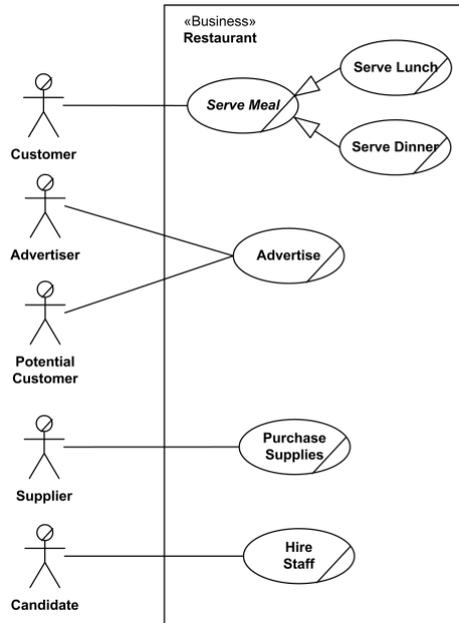
(a.)



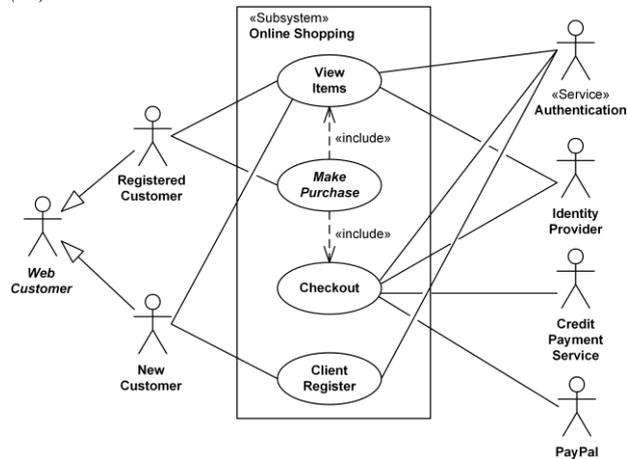
(b.) Restaurant – von außen



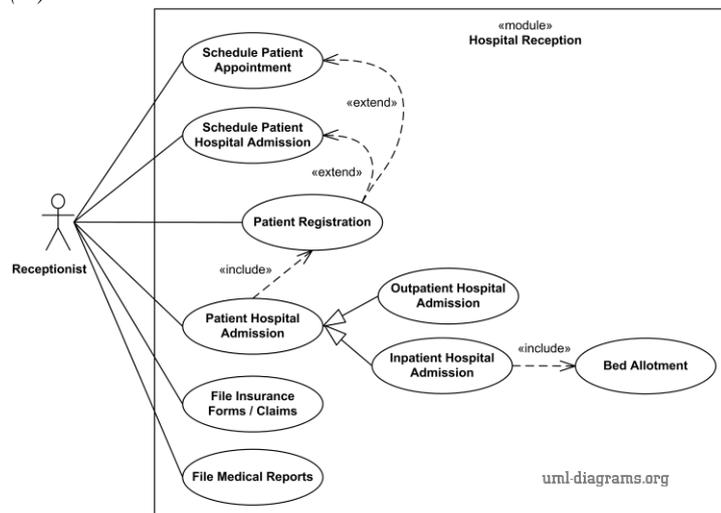
(c.) Restaurant – von innen

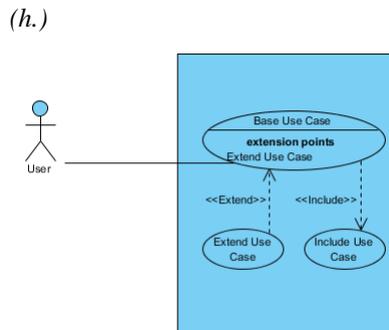
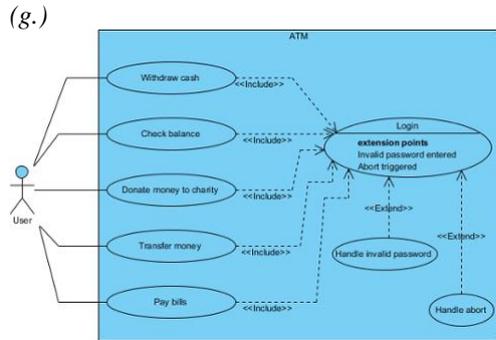
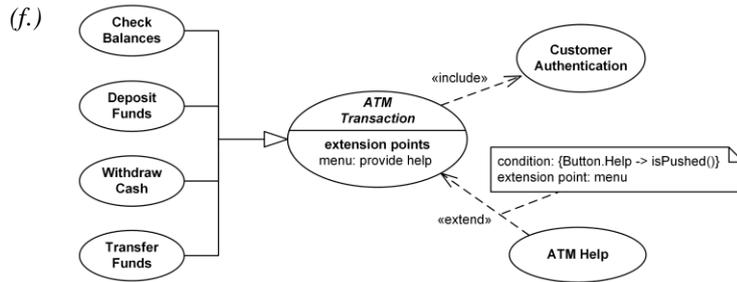


(d.)



(e.)





(3.) Zeichnen Sie für ein Fahrkartensystem das Anwendungsfalldiagramm!

Es gibt einen Akteur: Reisender

Es gibt die Fälle „Monatticket kaufen“, „Wochenticket kaufen“ und „Einmalticket kaufen“

Bei **allen** diesen Fällen **können** die Fehlerfälle „Zeitüberschreitung“, „Kauf abgebrochen“, „Automat kann nicht wechseln“ und „Automat hat kein Papier mehr“ auftreten!

(Also include the following exceptional cases: Time-Out (i.e., traveler took too long to insert the right amount), TransactionAborted (i.e., traveler selected the cancel button without completing the transaction), DistributorOutOfChange, and DistributorOutOfPaper.)

(4.) Ein einer Firma gibt es das alte DV-System parallel zu dem neuen DV-System.

Im alten System kann der Kunde eine Online-Bestellung durchführen, genauso wie im neuen System, wo er allerdings seine Kundendaten einsehen kann!

Der Mitarbeiter kann im alten und neuen System dem Kunden ein Angebot unterbreiten, im neuen System kann er zusätzlich Kunden neu anlegen, ändern und löschen!

Der Chef der Firma kann endlich im neuen System einen Verkaufsbericht erstellen und auf dem Drucker ausdrucken!

(5.) Projektverwaltung einer Firma:

Der Verwalter kann Rechnungen erstellen, Angebote erstellen, Aufträge eintragen und Auftragsbestätigungen erstellen.

Der Mitarbeiter trägt seine Arbeitszeit ein und das mit Hilfe eines Stundenzettels. Ein spezieller Mitarbeiter ist der Admin, der dann die Mitarbeiter verwalten kann.

(6.) Zu einem Konto gehören eine Kontonummer und ein Kontostand. Auf das Konto kann Geld eingezahlt werden. Unter der Bedingung, dass der Kontostand immer mindestens 1€ beträgt, kann von dem Konto auch Geld abgeboben werden. Von einem Konto kann Geld auf ein anderes Konto überwiesen werden. Eine Überweisung beinhaltet immer das Abheben eines Geldbetrages vom eigenen Konto und das Einzahlen auf ein anderes Konto.

(a.) Stellen Sie die oben beschriebene Klasse Konto als Klassendiagramm dar. Die Klasse soll einen Konstruktor besitzen, welcher bei der Objekterzeugung den Attributen frei wählbare Werte zuweisen kann. Für die Attribute müssen get-Methoden existieren, auf set-Methoden darf verzichtet werden.

(b.) Erstellen Sie ein Anwendungsfalldiagramm, welches alle im Einführungstext beschriebenen Anwendungsfälle enthält.

(c.) Erstellen Sie den Javacode der Klasse Konto!

(d.) Stellen Sie den folgenden Ablauf in einem Sequenzdiagramm dar:

Es existieren zwei Konto-Objekte, k1 und k2. Ein Kunde überweist vom Konto k1 200€ auf das Konto k2. Die Überweisung kann durchgeführt werden, weil sich genügend Geld auf dem Konto k1 befindet.

(e.) Erstellen Sie eine Klasse Bank, welche die Main-Methode enthält. In der Main-Methode werden zwei Konto-Objekte k1 (Kontonummer 123) und k2 (Kontonummer 456) erzeugt. Bei der Erzeugung werden beide Kontostände auf 300€ gesetzt. Anschließend soll der in d) beschriebene Ablauf durchgeführt werden. Zur Kontrolle sind anschließend beide Kontostände auszugeben.

(7.) 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.

(8.) 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.

UML - Anwendungsfalldiagramme - Lösungen

(1.) In einem Anwendungsfalldiagramm werden Anwendungsfälle (use cases) und Akteure mit ihren Abhängigkeiten und Beziehungen dargestellt.

Ein Anwendungsfall beschreibt eine Funktionalität des Softwaresystems.

– Ein Akteur ist eine Rolle, die ein Benutzer des Softwaresystems spielt.

– Eine Linie zwischen Akteur und Anwendungsfall bedeutet, dass eine Kommunikation stattfindet.

Mit Hilfe der extend-Beziehung wird ein Anwendungsfall durch einen anderen erweitert (kann-Beziehung). Die include-Beziehung ermöglicht es, dass die gemeinsame Funktionalität mehrerer Anwendungsfälle durch einen weiteren Anwendungsfall beschrieben wird (muss-Beziehung).

(2.)

(a.) Akteure sind hier die Passagiere und zwar speziell der Touristenführer, die minderjährigen und die Passagiere mit besonderen Bedürfnissen. Die Minderjährigen und die Passagiere mit besonderen Bedürfnissen sind eine Untergruppe (da abgeleitet) von den Passagieren und verhalten sich genauso

Die erste „Kommunikation“ findet zwischen dem Touristenführer und dem Gruppen-Check-In statt. Bei Passagieren ist dies der individuelle Check-In, bei dem Sie auch die Security passieren müssen.

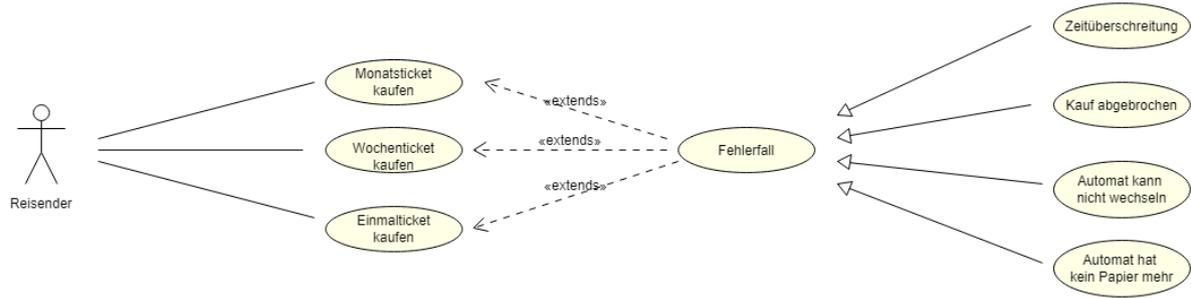
Anwendungsfälle sind der Gruppen-Check-In, der individuelle Check-In, das Security Screening usw.

Mit Hilfe der extend-Beziehung wird ein Anwendungsfall „Individual Check-In“ durch den „Baggage Check-In“ erweitert (kann-Beziehung). Dies macht Sinn, da es Passagiere ohne Gepäck gibt, die den „Baggage-Check-In“ dann nicht benötigen, so dass dieser bei dem Fall „Individual-Check-In“ optional ist. Analog ist das „Baggage Handling“ bei den Fällen „Counter-Check-In“ sowie „Kiosk-Check-In“ für Passagiere ohne Gepäck optional; daher wird das „Baggage Handling“ in diesen Fällen als „extend“ gekennzeichnet!

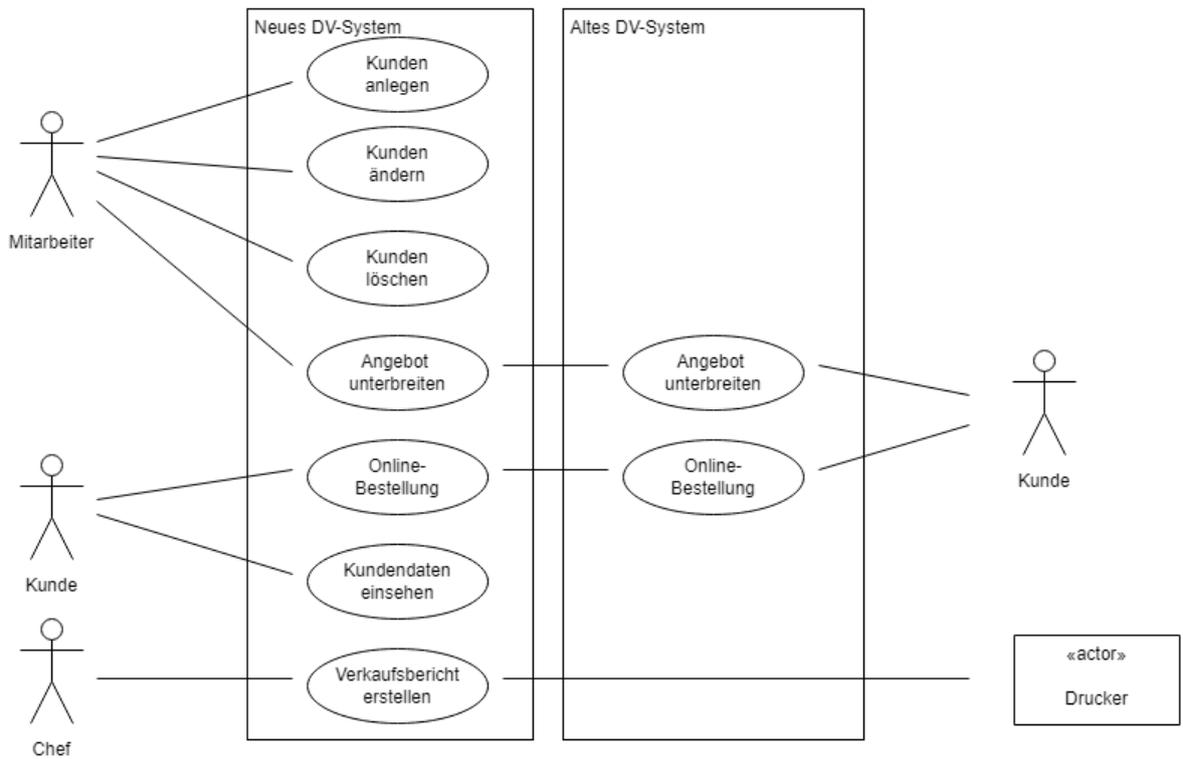
Die include-Beziehung ermöglicht es, dass die gemeinsame Funktionalität mehrerer Anwendungsfälle durch einen weiteren Anwendungsfall beschrieben wird (muss-Beziehung). Hier: Der Anwendungsfall „Group Check-In“ benötigt „Individual Check-In“, da die einzelnen Personen in der Gruppe einchecken müssen!

(b.) und folgende - ohne Lösung!

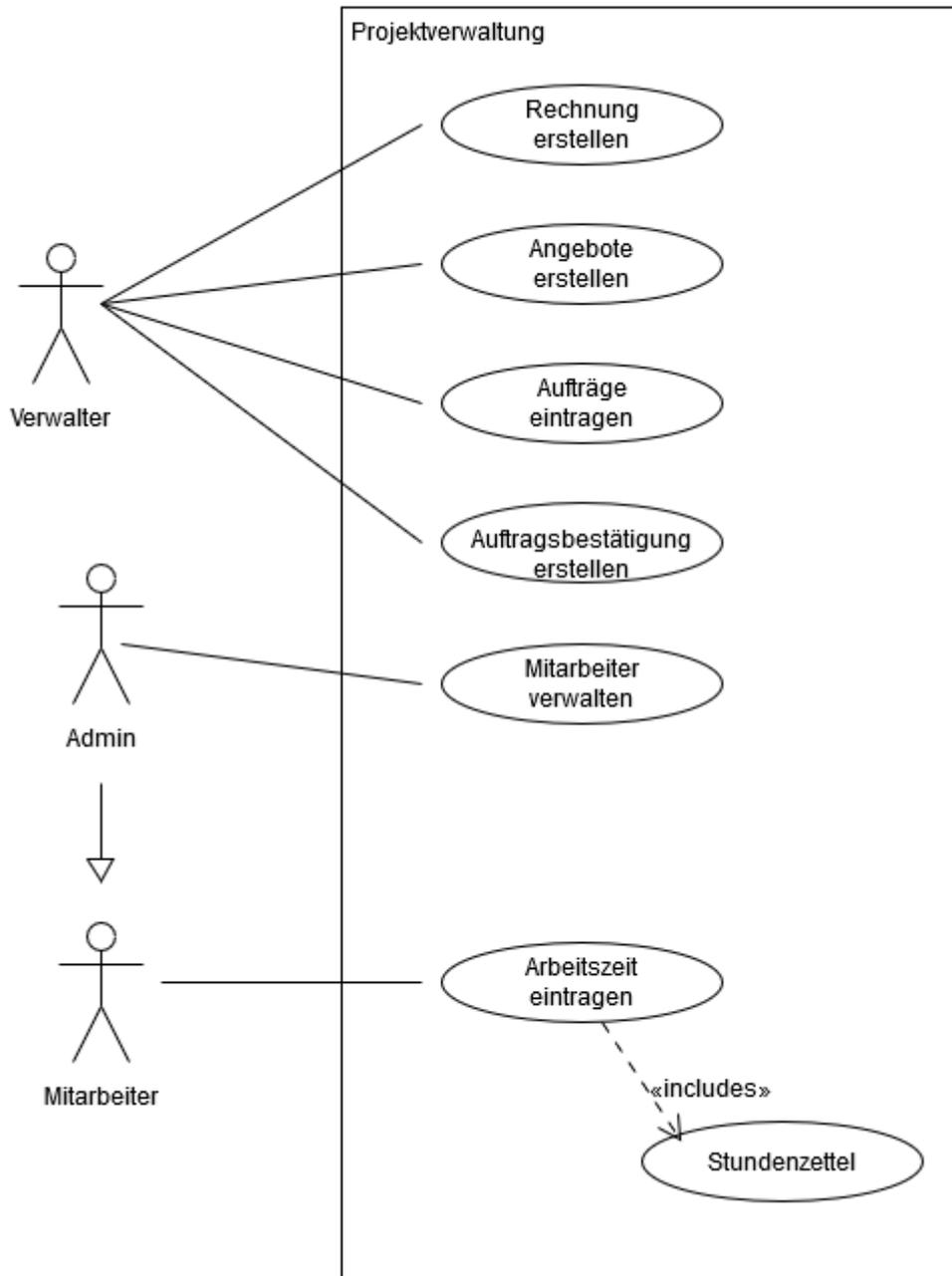
(3.)



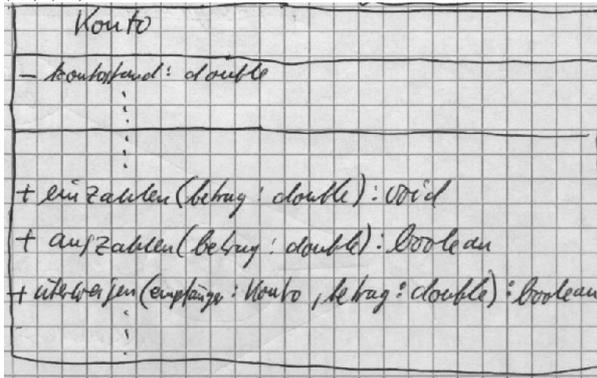
(4.)



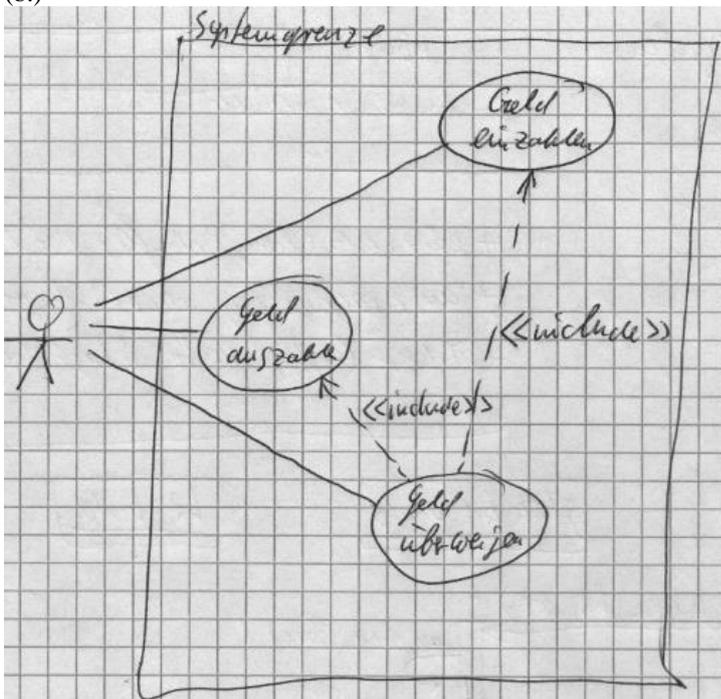
(5.)



(6.) (a.)

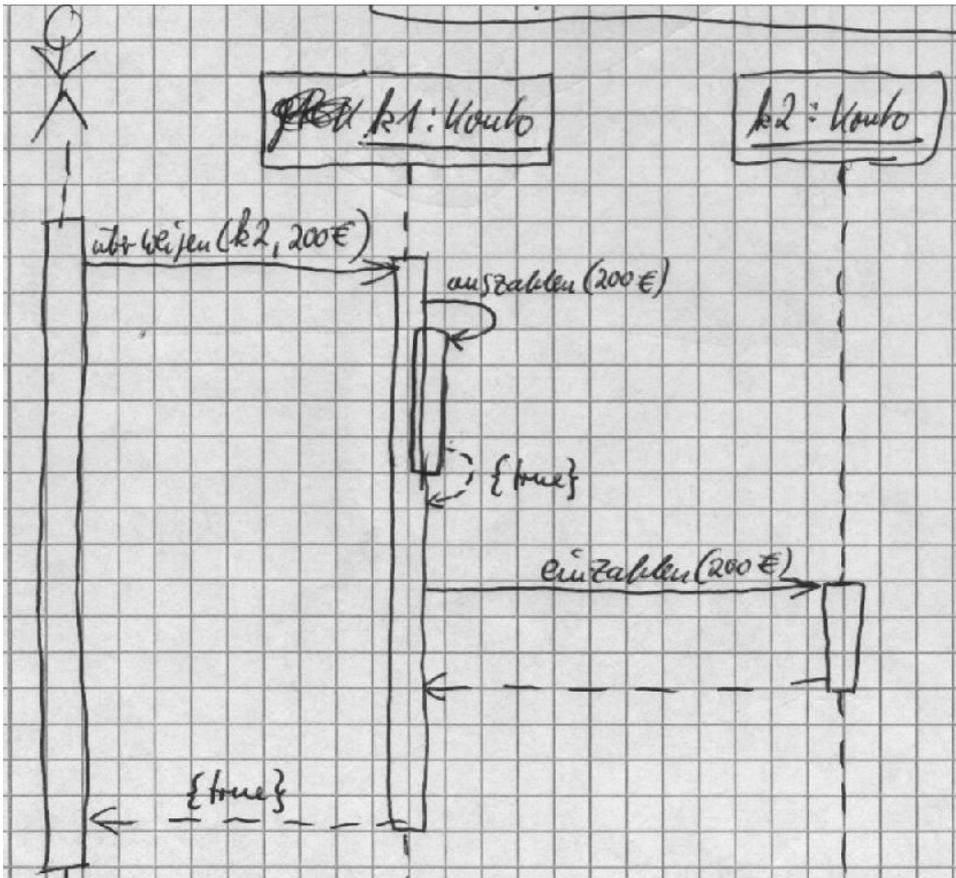


(b.)



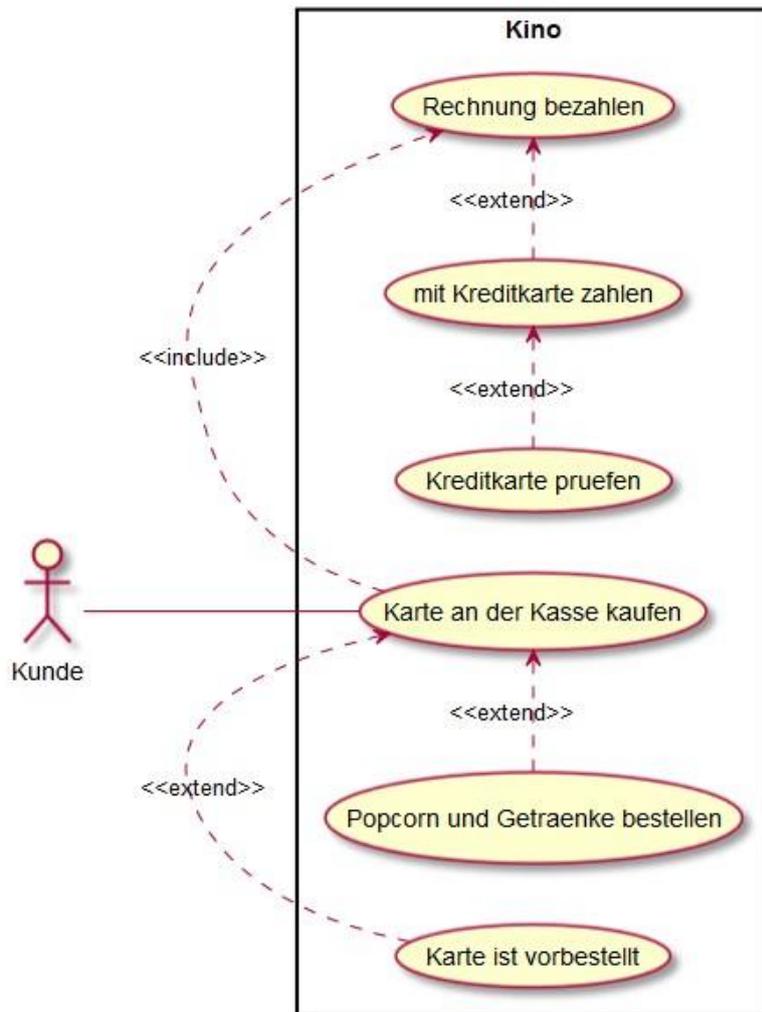
(c.) als Dateiversion

(d.)



(e.) als Dateiversion

(7.)



(8.)

