1. Anwendungsfall-Diagramm Statistikabfragen

Die Soft GmbH wurde von der FAQ GmbH mit der Entwicklung einer Software beauftragt, die den Kunden der FAQ GmbH einen Onlinezugang zu statistischen Daten ermöglicht. Folgende Anforderungen an die Software "Statistikabfragen" liegen vor:

- Jeder Nutzer des Onlineangebotes der FAQ GmbH kann Standardstatistiken abrufen.

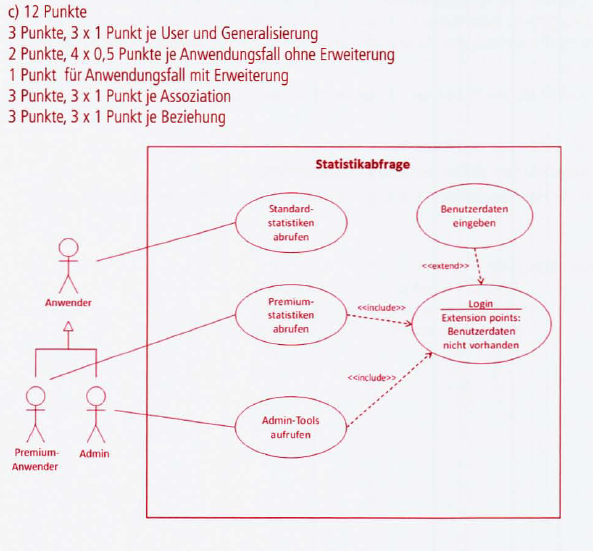
- Ein Premiumnutzer kann zusätzlich Premiumstatistiken abrufen. Dazu ist ein Login erforderlich.

- Falls die Login-Daten nicht vorliegen (z.B. Erstanmeldung), muss dieser Nutzer die erforderlichen Daten eingeben.

- Ein Administrator kann verschiedene Admin-Tools abrufen. Auch dazu ist ein Login erforderlich.

- Ein Administrator kann nur Standardstatistiken abrufen.

Erstellen Sie anhand der vorliegenden Informationen ein UML-Anwendungsfalldiagramm für die Software „Statistikabfraqen".



GA1 FIA 2014 Winter

Der extension Point muss als Notiz noch am <<extend>> - Pfeil gemeinsam mit der Bedingung (condition) angegeben werden – siehe Notation.

1. Anwendungsfall-Diagramm Webshop

Die KLÜBERO GmbH soll den Webshop der Media AG entwickeln.

In einer ersten Besprechung wurden einige grundlegende Anwendungsfälle für den Webshop herausgearbeitet.

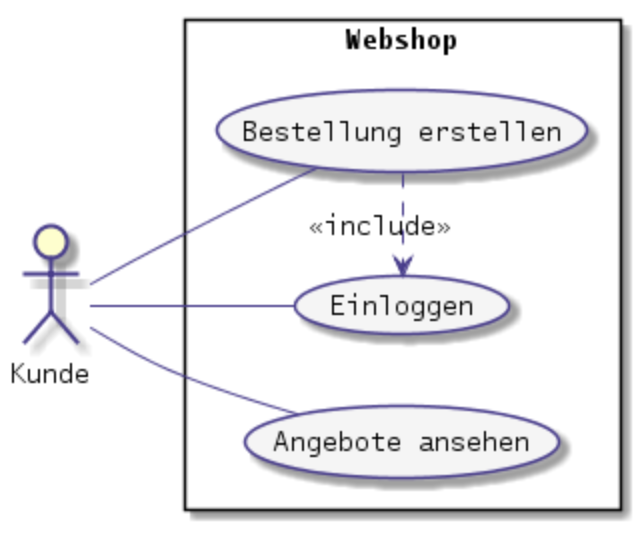
Der Kunde soll

- sich die Angebote des Webshops ansehen können.

- sich beim Webshop einloggen können.

- eine Bestellung aufgeben können. Dieses kann nur geschehen, wenn sich der Kunde vorher im Webshop eingeloggt hat und erweitert somit den Anwendungsfall des Einloggens.

Entwickeln Sie ein UML-Anwendungsfalldiagramm, welches die gegebenen Sachverhalte darstellt.



|  |  |
| --- | --- |
| Hinweis: die nebenstehende Lösung der IHK ist aus meine Sicht Unsinn – Einloggen wird inkludiert – die Fragestellung mit „erweitern“ macht daher wenig Sinn... GA2 2016/17 Winter |  |

1. Datenerfassung für eine Kundendatei

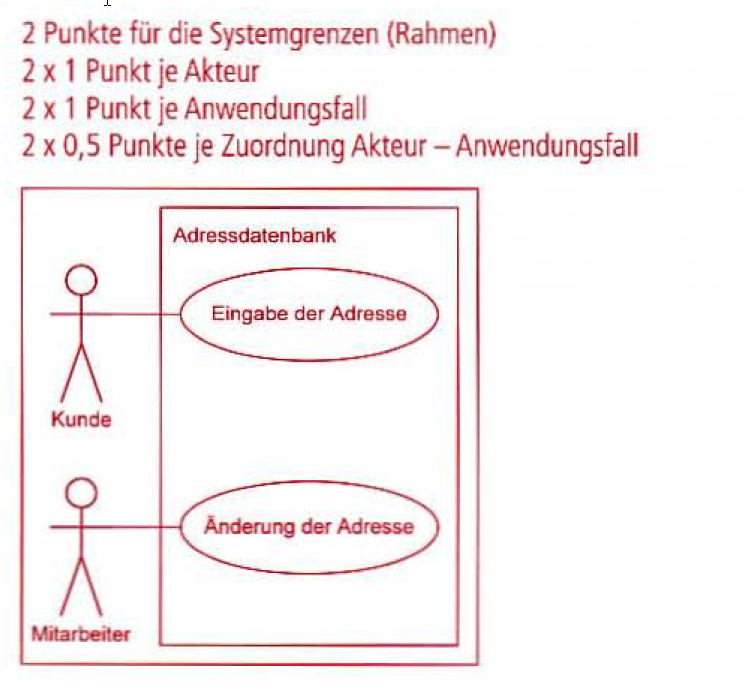
Die Media AG legt Wert auf eine aktuelle Kundendatei ihres Webshops und ändert Adressen, sobald sie von der Adressänderung eines Kunden erfährt. Für die Kundendatei des Webshop der Media AG ist für die Adresserfassung und –pflege folgender Anwendungsfall geplant:

- Ein Neukunde gibt seine Adressdaten in die Kundendatei ein.

- Ein Mitarbeiter ändert Adressdaten.

Stellen Sie den Grundaufbau eines UML-Anwendungsfalldiagramms „Datenerfassunq" dar.

Sommer 2013 GAII



1. Anwendungsfälle für CRM-System

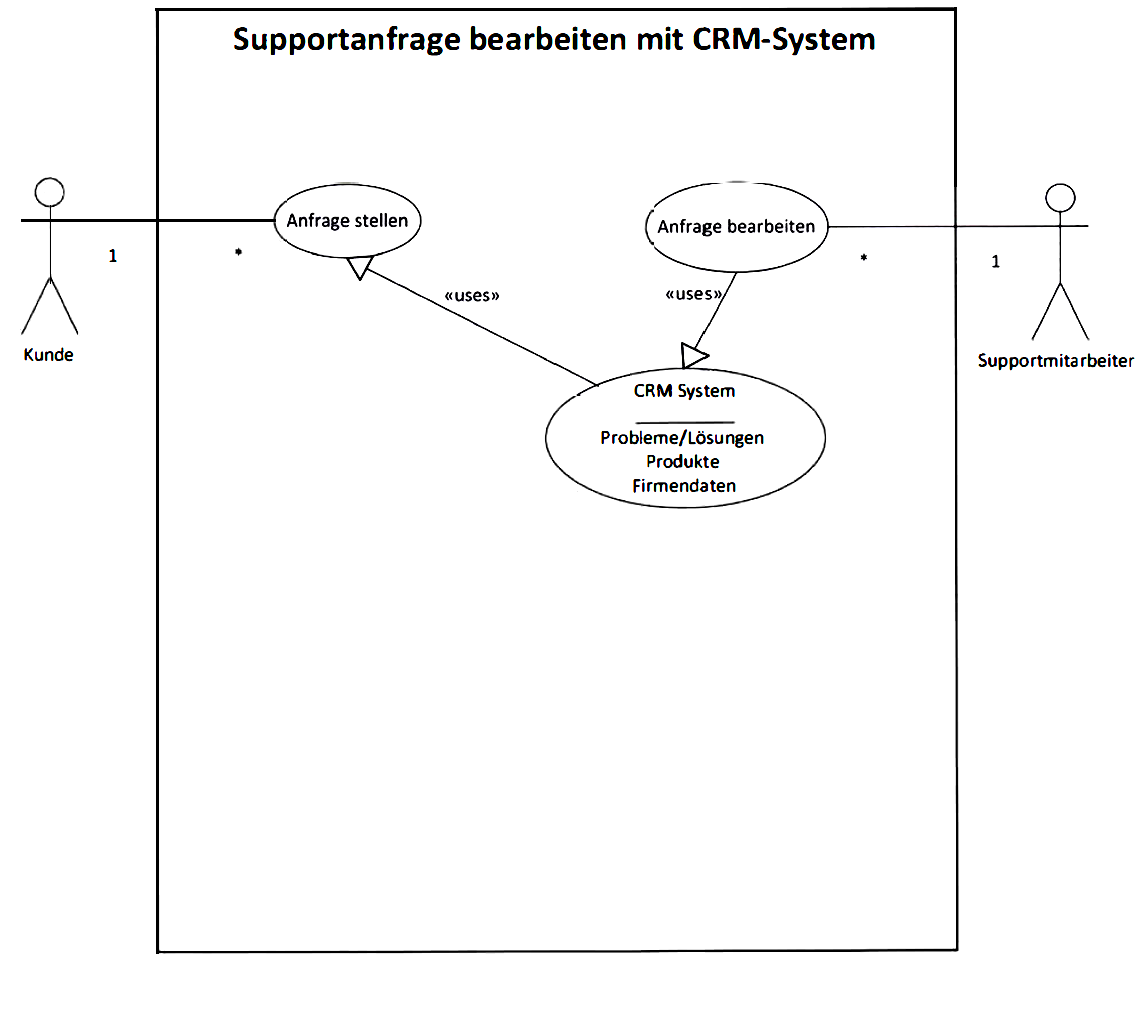
Die Glabal KG erwägt, ein CRM-System einzuführen (CRM = Customer Relationship Management) und will von der Easy IT GmbH beraten werden.

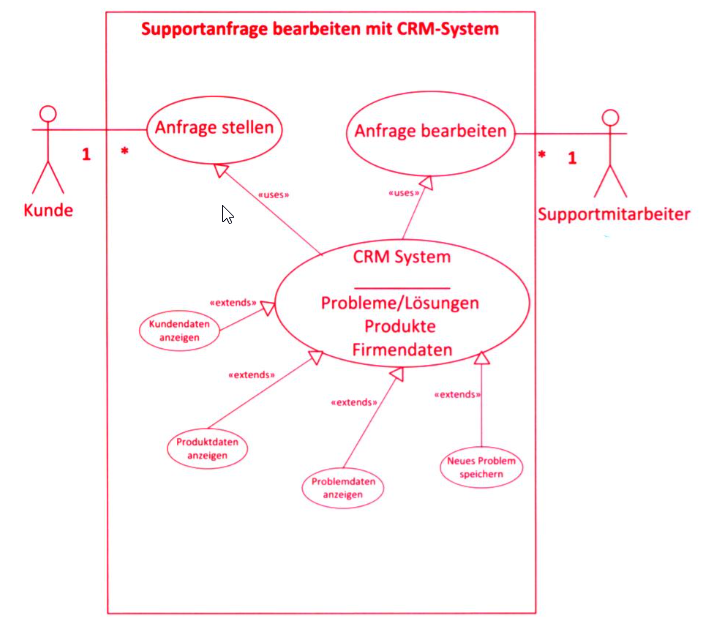
a) Mit dem CRM-System soll ein Supportmitarbeiter die Supportanfragen von Kunden wie folgt bearbeiten: Der Supportmitarbeiter soll die Datenbank öffnen und daraus folgende Daten erfassen:

- 1. Firmendaten, 2. gekaufte Produkte und 3. bisher aufgetretene Probleme.

Er soll weiterhin das aktuelle Problem in der Datenbank des CRM-Systems protokollieren.

Wie der Supportmitarbeiter die oben dargestellte Supportanfrage eines Kunden mithilfe des CRM-Systems bearbeitet, soll in einem Use-Case-Diagramm abgebildet werden. Vervollständigen Sie dazu den folgenden Entwurf.





FISy GA1 Winter 2010-11

1. Reservierungen

Sie sind Mitarbeiter/-in der Krake AG. Die Krake AG ist ein Vermieter von Baggern und anderen Großgeräten für das Baugewerbe. Die betriebseigene IT-Abteilung soll im Rahmen eines Projekts Geschäftsprozesse verbessern.

Ein Ziel des Projektes ist die Verbesserung des Kundenservice. Im Rahmen dieses Vorhabens soll eine neue Internetpräsenz erstellt werden.

Die neue Internetpräsenz soll unter anderem folgende Funktionalität bieten:

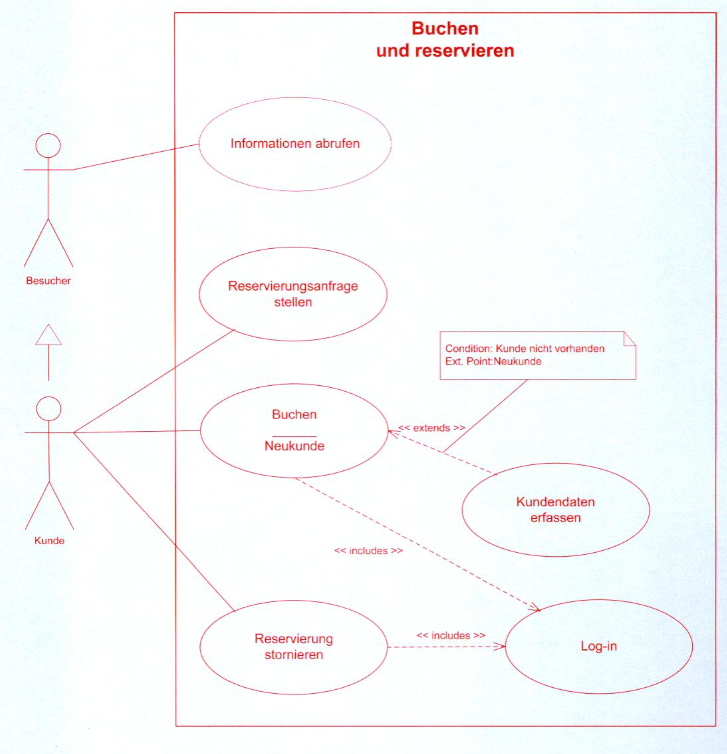
Alle Besucher der Webseite können allgemeine Informationen der Krake AG abrufen (Leistungs-und Produktportfolio und AGB).

Ein Kunde kann Reservierungsanfragen stellen sowie Buchungen und Stornierungen durchführen. Von einem Neukunden werden zunächst die Kundendaten erfasst. Für Buchungen und Stornierungen muss sich ein Kunde einloggen; seine eingegebenen Daten werden überprüft.

Erstellen Sie ein entsprechendes Anwendungsfalldiagramm.

FIA GA1 Sommer 2010

extend und include ohne „s“!!!



1. Anwendungsfall-Diagramm Kurierfahrten (10P)

Die Mustermann GmbH erhält von der Kurier GmbH folgende Beschreibung von den bei einer Kurierfahrt auszuführenden Tätigkeiten.

- Erfassung der Daten einer Sendung (u. a. Gewicht)

- Transport der Sendung innerhalb des Stadtgebiets von Bielefeld (Tür zu Tür)

- Persönliche Übergabe der Sendung und Quittierung des Empfangs durch Empfänger, sofern dies vom Auftraggeber gewünscht ist

- Automatische Rechnungserstellung nach Abschluss der Kurierfahrt

Entwickeln Sie ein Use-Case-Diagramm (Anwendungsfall-Diagramm), das die oben beschriebene Auftragsdurchführung darstellt.

|  |  |  |
| --- | --- | --- |
| Beispiellösung 1: | | @startuml  left to right direction  actor :Kurier:  actor :Kunde:  rectangle Kurierfahrt {  usecase erfassen as "Erfassen der Daten einer Sendung"  usecase transport as "Transport der Sendung innerhalb von Bielefeld  --  <i>extension points:</i>  Quittierung  "  usecase quittierung as "Persönliche Übergabe und Quittierung der Sendungsannahme"  transport <.quittierung : <<extend>>  note top on link  <i>condition:</i>  {Kunde wünscht Quittierung}  <i>extension point:</i>  Quittierung  end note  usecase rechnung as "Automatische Rechnungserstellung"  transport .> erfassen : <<include>>  transport ..> rechnung : <<include>>  Kurier -- transport  Kurier -- erfassen  Kurier -- rechnung  Kurier -- quittierung  Kunde -- quittierung  } |
| Beispiellösung 2: | | @startuml  skinparam DefaultFontName "Lucida Sans Typewriter"  skinparam UseCase{  BorderColor DarkSlateBlue  BorderColor DarkSlateBlue  }  skinparam Note{  BorderColor DarkSlateBlue  BackgroundColor LightYellow  }  skinparam ArrowColor DarkSlateBlue  skinparam ActorBorderColor DarkSlateBlue  skinparam UseCaseBackgroundColor whitesmoke  left to right direction  actor :Kurier:  actor :Kunde:  rectangle Kurierfahrt {  usecase uebergabe as "Übergabe  --  <i>extension points:</i>  Quittierung  "  Kurier -- uebergabe  usecase quittierung as "Quittierung durch den Kunden"  uebergabe <.left. quittierung : <<extend>>  note top on link  <i>condition:</i>  {Kunde wünscht Quittierung}  <i>extension point:</i>  Quittierung  end note  Kunde -up- quittierung  Kunde -up- uebergabe  usecase rechnung as "Automatische Rechnungserstellung"  usecase transport as "Transport der Sendung"  usecase erfassen as "Erfassen der Daten einer Sendung"  transport .right.> rechnung : <<include>>  transport ..> erfassen : <<include>>  transport .left.> uebergabe : <<include>>  Kurier -- erfassen  Kurier -- transport  Kurier -- rechnung  Kurier - quittierung  }  @enduml |
| In der IHK-Beispiellösung sieht es wie folgt aus: GA1 FIA 2009 Sommer   * Systemgrenze und Name fehlt. * Stereotypen falsch bezeichnet (extendS) und ohne << >> * extensionpoint und condition fehlen | Bewertungsvorschlag der IHK:   * je Akteur 1Ptk, je Anwendungsfall 1 Pkt. * include und extend je 2 Pkt | |

Das Mitarbeiterverwaltungssystem soll folgende Funktionalität bieten:

- Jeder Mitarbeiter/jede Mitarbeiterin kann sich den Status eines Lesegerätes anzeigen lassen.

- Jeder Mitarbeiter/jede Mitarbeiterin kann sich die Protokolldaten zu einem Lesegerät anzeigen lassen.

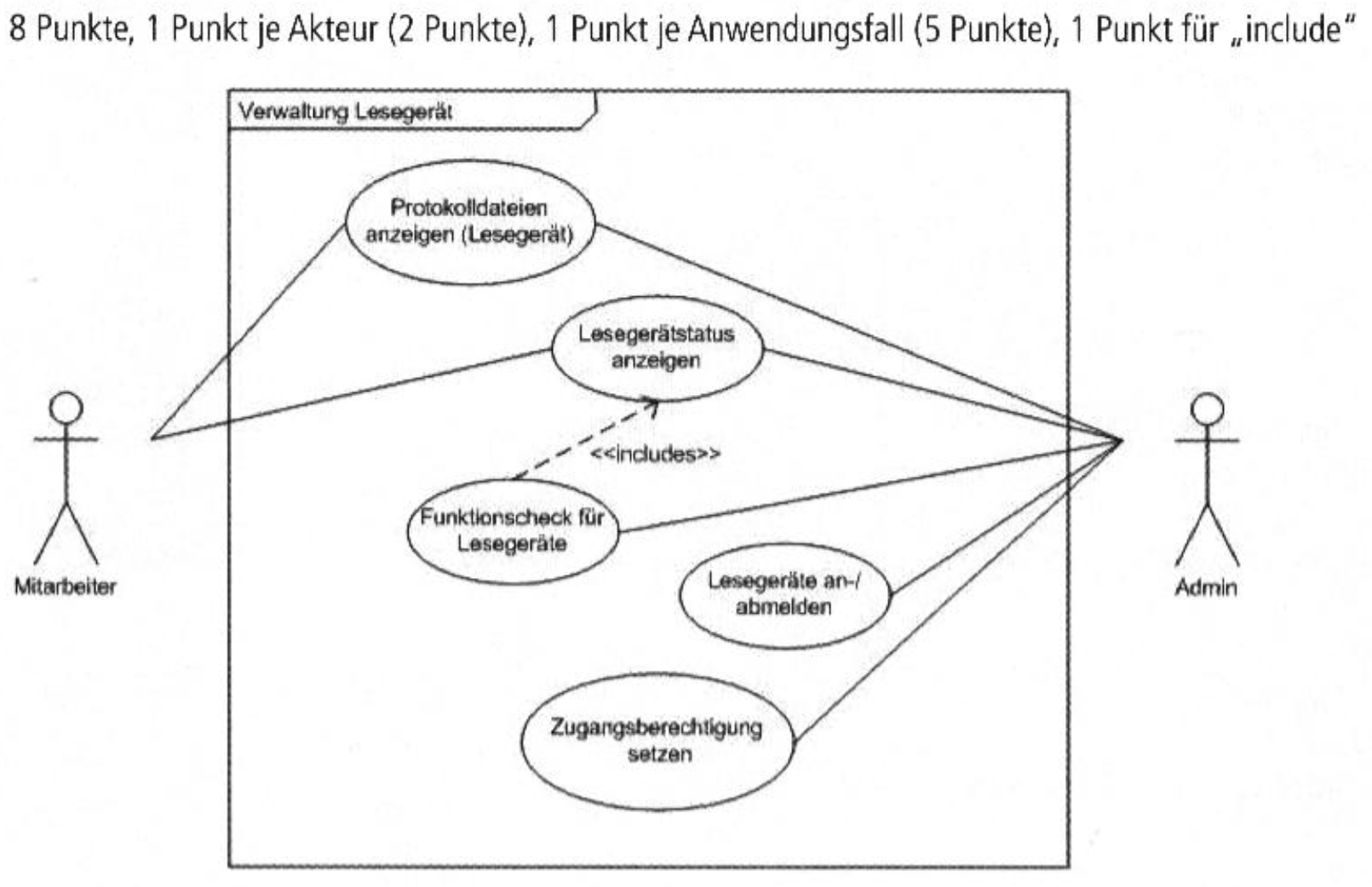
- Ein Administrator kann Lesegeräte anmelden und abmelden.

- Ein Administrator kann Zugangsberechtigungen an einem Lesegerät setzen.

- Ein Administrator kann einen Funktionscheck für ein Lesegerät durchführen. In diesem Fall wird stets der Status des Lesegerätes angezeigt.

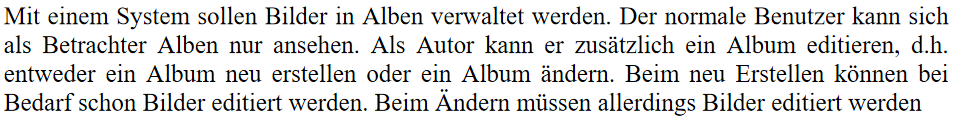
Jeder Administrator ist ein Mitarbeiter/eine Mitarbeiterin.

Erstellen Sie ein Use-Case-Diagramm.

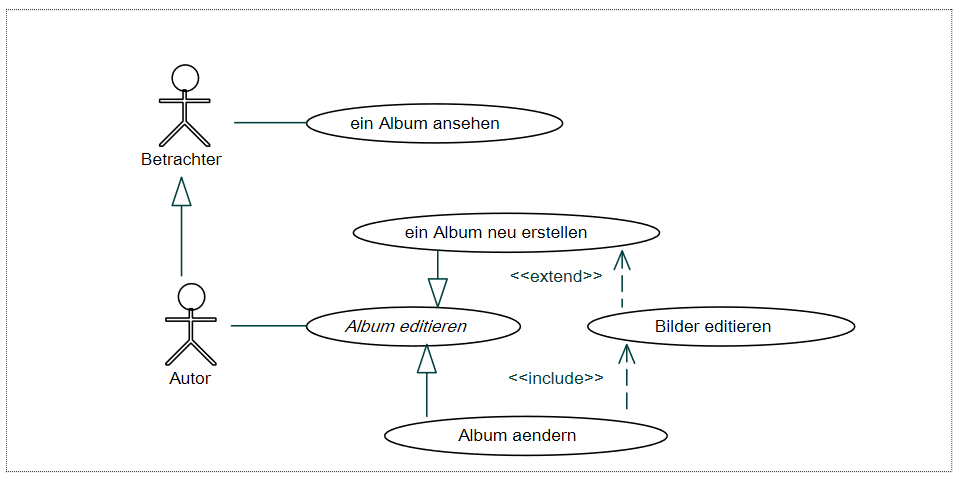


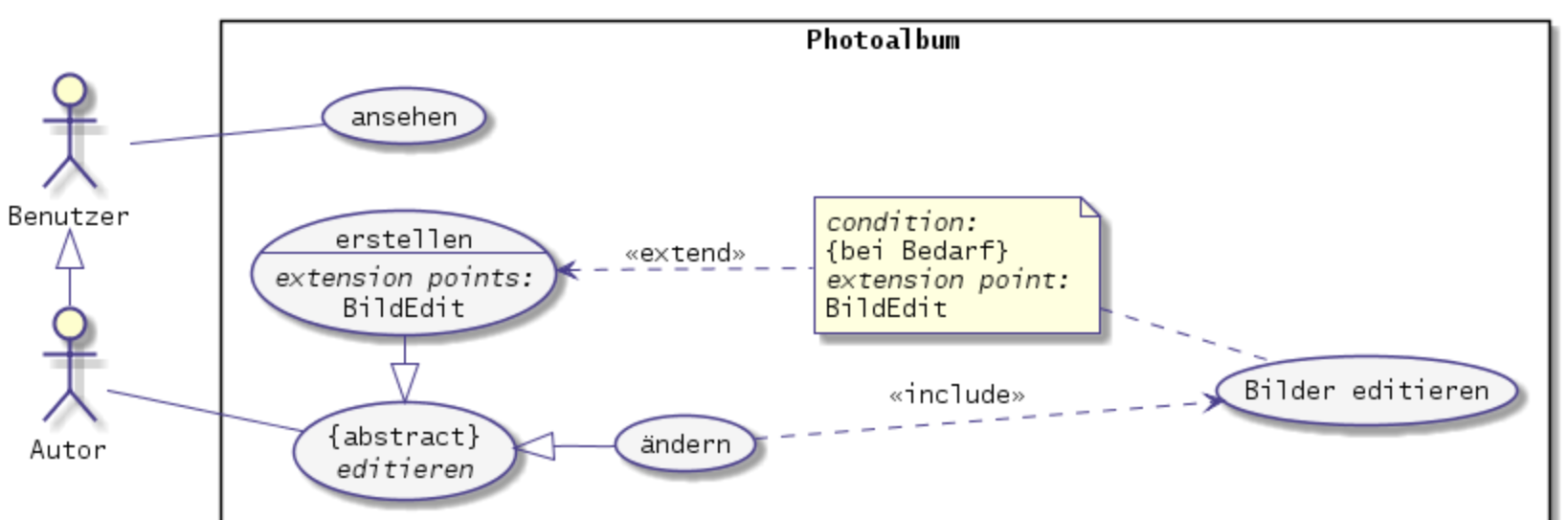
GA1 FIA 2008W

1. Anwendungsfall-Diagramm Fotoalben

Erstellen Sie ein Use-Case-Diagramm, das die folgende Funktionalität spezifiziert:

http://klaus.hdm-stuttgart.de/data/13037\_Informatik\_4/13037-WS2006-Ihler-279.pdf

Die in der Orginalquelle hinterlegte Lösung ist unvollständig.



@startuml

skinparam DefaultFontName "Lucida Sans Typewriter"

skinparam UseCase{

BorderColor DarkSlateBlue

BorderColor DarkSlateBlue

}

skinparam Note{

BorderColor DarkSlateBlue

BackgroundColor LightYellow

}

skinparam ArrowColor DarkSlateBlue

skinparam ActorBorderColor DarkSlateBlue

skinparam UseCaseBackgroundColor whitesmoke

left to right direction

actor :Benutzer: as user

actor :Autor: as author

user <|-right- author

rectangle Photoalbum {

usecase (Album ansehen) as see

usecase ({abstract}\n<i>Album editieren</i>) as edit

usecase create as "Album erstellen

--

<i>extension points:</i>

BildEdit"

usecase (Album ändern) as change

usecase (Bilder editieren) as picedit

user -- see

author -- edit

edit <|-left- create

edit <|-right- change

change .down.> picedit : <<include>>

note as cond

<i>condition:</i>

{bei Bedarf}

<i>extension point:</i>

BildEdit

end note

create <.down. cond : <<extend>>

cond .down. picedit

@enduml

1. Anwendungsfall-Diagramm Vorlesungen

a) Ein Student und seine Freunde (zwei Rollen) besuchen eine Vorlesung. Zeichnen Sie ein einfaches Use Case-Diagramm zu dieser Situation.

|  |  |
| --- | --- |
| cid:image001.png@01D3D0D9.9847CF80 | ‘Quelltext für PlantUML : <https://www.planttext.com/>  @startuml  left to right direction  skinparam packageStyle rectangle  actor Student AS student  actor Freund as freund  rectangle Uni {    student "1" -- (Vorlesung besuchen)    freund "1..\*" -- (Vorlesung besuchen)  }  @enduml |

b) Um die Vorlesung besuchen zu können, muss der Student und seine Freunde die Wohnung verlassen. Erweitern Sie das Diagramm um diesen und noch um 3 weitere Anwendungsfälle, die sinnvolle Erweiterungen darstellen.

|  |  |
| --- | --- |
|  | ‘Quelltext für PlantUML : <https://www.planttext.com/>  @startuml  left to right direction  skinparam packageStyle rectangle  actor Student AS student  actor Freund as freund  rectangle Uni {  usecase vorlesung as "Vorlesung besuchen"  student "1" -- vorlesung  freund "1..\*" -- vorlesung  vorlesung .left.> (Haus verlassen) : <<include>>  vorlesung ..> (Unterlagen einpacken) : <<include>>  vorlesung ..> (offene Fragen notieren) : <<include>>  }@enduml |

c) Es gibt zwei spezielle Arten die Vorlesung zu besuchen: sich als Gruppe zusammenzusetzen oder sich jeweils einzeln in die Vorlesung zu setzen. Erweitern Sie das Diagramm mittels Vererbungsbeziehung um diesen Anwendungsfall.

|  |  |
| --- | --- |
|  | ‘Quelltext für PlantUML : <https://www.planttext.com/>  left to right direction  skinparam packageStyle rectangle  actor Student AS student  actor Freund as freund  rectangle Uni {  usecase vorlesung as "Vorlesung besuchen"  student "1" -- vorlesung  freund "1..\*" -- vorlesung  vorlesung .left.> (Haus verlassen) : <<include>>  vorlesung ..> (Unterlagen einpacken) : <<include>>  vorlesung ..> (offene Fragen notieren) : <<include>>    usecase (Vorlesung in Gruppe gemeinsam hören) as vorlesunggemeinsam  usecase (Vorlesung einzeln anhören) as vorlesungeinzeln  vorlesung <|- vorlesunggemeinsam  vorlesung <|-- vorlesungeinzeln  } |

c) Eine Anwendungsfall in Ihrem Diagramm kann als „abstract“ deklariert werden. Tragen Sie Ihre Wahl in das Diagramm ein.

|  |  |
| --- | --- |
| (Vorlesung besuchen ist abstrakt => kursiv) | ‘Quelltext für PlantUML : <https://www.planttext.com/>  left to right direction  skinparam packageStyle rectangle  actor Student AS student  actor Freund as freund  rectangle Uni {  usecase vorlesung as "<I>Vorlesung besuchen</I>"  student "1" -- vorlesung  freund "1..\*" -- vorlesung  vorlesung .left.> (Haus verlassen) : <<include>>  vorlesung ..> (Unterlagen einpacken) : <<include>>  vorlesung ..> (offene Fragen notieren) : <<include>>    usecase (Vorlesung in Gruppe gemeinsam hören) as vorlesunggemeinsam  usecase (Vorlesung einzeln anhören) as vorlesungeinzeln  vorlesung <|- vorlesunggemeinsam  vorlesung <|-- vorlesungeinzeln  } |

d) Wenn der Student seine Wohnung verlässt, wird er mit dem Auto fahren, sofern das möglich ist. Erweitern Sie das Diagramm mittels extend-Beziehung um diesen Anwendungsfall.

|  |  |
| --- | --- |
| (Vorlesung besuchen ist abstrakt => kursiv) | ‘Quelltext für PlantUML : <https://www.planttext.com/>  @startuml  skinparam DefaultFontName "Lucida Sans Typewriter"  skinparam UseCase{  BorderColor DarkSlateBlue  BorderColor DarkSlateBlue  }  skinparam Note{  BorderColor DarkSlateBlue  BackgroundColor LightYellow  }  skinparam ArrowColor DarkSlateBlue  skinparam ActorBorderColor DarkSlateBlue  skinparam UseCaseBackgroundColor whitesmoke  skinparam packageStyle rectangle  left to right direction  actor Student AS student  actor Freund as freund  rectangle Uni {  usecase vorlesung as "<I>Vorlesung besuchen</I>  --  <i>extension points:</i>  AutoMöglich"  student "1" -- vorlesung  freund "1..\*" -- vorlesung  vorlesung .left.> (Haus verlassen) : <<include>>  vorlesung ..> (Unterlagen einpacken) : <<include>>  vorlesung ..> (offene Fragen notieren) : <<include>>    usecase (Vorlesung in Gruppe gemeinsam hören) as vorlesunggemeinsam  usecase (Vorlesung einzeln anhören) as vorlesungeinzeln  vorlesung <|-- vorlesunggemeinsam  vorlesung <|-- vorlesungeinzeln  vorlesung <. (Autofahren) : <<extend>>  note top on link  <i>condition:</i>  {wenn Autofahren möglich ist}  <i>extension point:</i>  AutoMöglich  end note  }  @enduml |

ursprünglich von: https://www.isf.cs.tu-bs.de/cms/teaching/2012w/se1/solution2.pdf

1. Anwendungsfall-Diagramm Kino

Entwerfen sie ein Use Case-Diagramm 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. Sofern mit Kreditkarte bezahlt wird, wird diese einer automatischen Prüfung unterzogen.

|  |  |
| --- | --- |
|  | ‘Quelltext für PlantUML : <https://www.planttext.com/>  @startuml  skinparam DefaultFontName "Lucida Sans Typewriter"  skinparam UseCase{  BorderColor DarkSlateBlue  BorderColor DarkSlateBlue  }  skinparam Note{  BorderColor DarkSlateBlue  BackgroundColor LightYellow  }  skinparam ArrowColor DarkSlateBlue  skinparam ActorBorderColor DarkSlateBlue  skinparam UseCaseBackgroundColor whitesmoke  skinparam packageStyle rectangle  left to right direction  actor Gast AS gast  rectangle Kinokasse {  usecase kartenkauf as "Karten an der Kasse kaufen  --  <i>extension points:</i>  Vorbestellung"  gast -- kartenkauf  usecase (Zurückgelegte Karte holen) as vorbestellung  kartenkauf <. vorbestellung : <<extend>>  note top on link  <i>condition:</i>  {Karte wurde reserviert}  <i>extension point:</i>  Vorbestellung  end note    usecase (Getränke kaufen) as getraenk  gast -- getraenk    usecase (Popcorn kaufen) as popcorn  gast -- popcorn    usecase rechnung as "Rechnung bezahlen  --  <i>extension points:</i>  Kreditkarte"  popcorn ..> rechnung : <<include>>  getraenk ..> rechnung : <<include>>  kartenkauf ..> rechnung : <<include>>    usecase (Kreditkarte prüfen) as kreditkarte  kreditkarte .left.> rechnung : <<extend>>  note top on link  <i>condition:</i>  {Kreditkartenkauf}  <i>extension point:</i>  Kreditkarte  end note  } |
| Lösung lt. Website... bitte Anmerkungen rechts beachten! | Diese Lösung hat einige Probleme:  - extend als bedingte Erweiterung macht nur unter Nennung der Condition / des Extention-Points wirklich Sinn  - eine allzu abenteuerliche verschachtelung von include und extend trägt nicht zur Übersicht bei  - Popcorn und Getränkekauf nur mit Kinokarte möglich?!? |

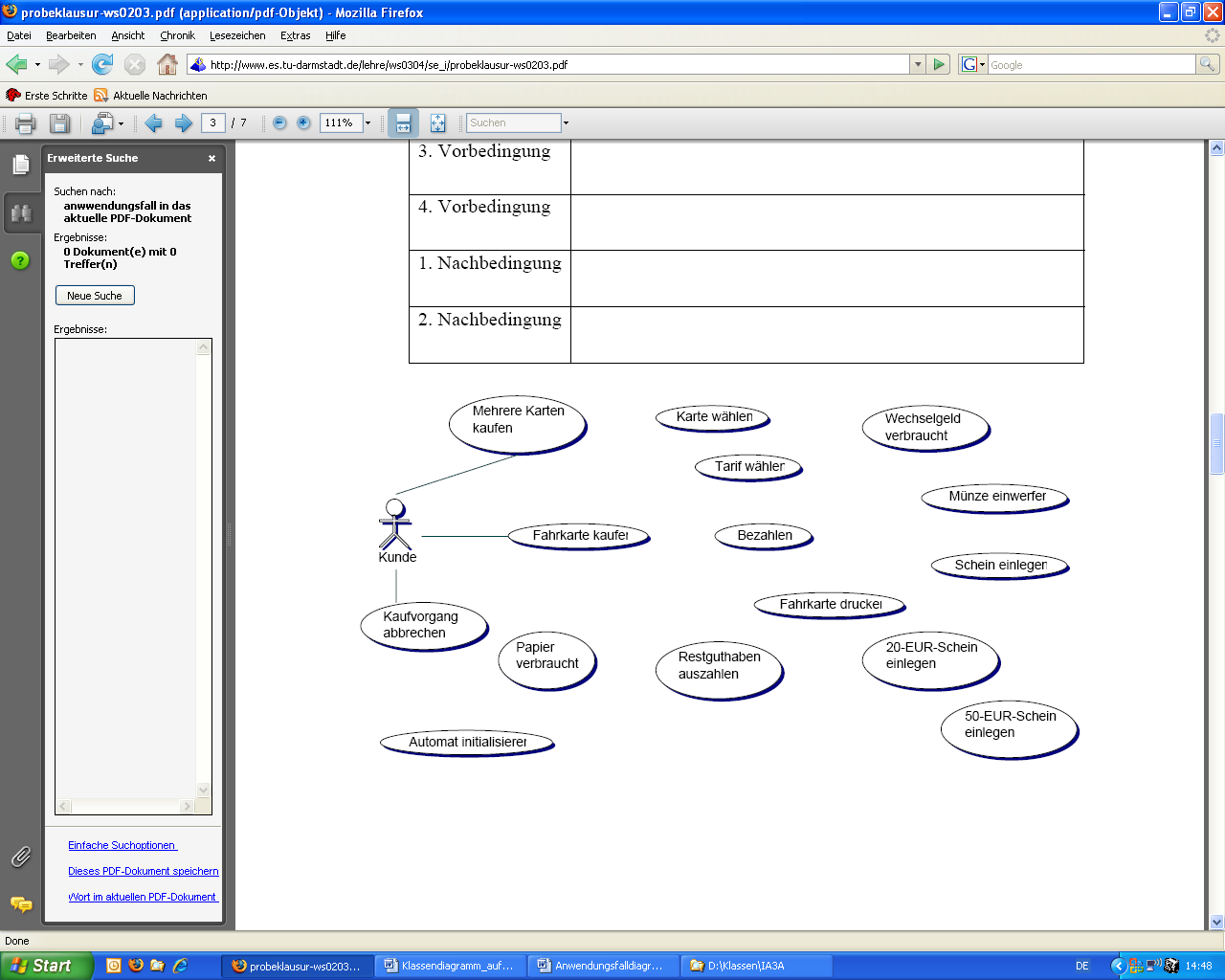
In Anlehnung an: https://www.isf.cs.tu-bs.de/cms/teaching/2012w/se1/solution2.pdf

1. Beschreibung der Aktivitäten eines Use-Case-Diagramms
2. Beschreiben Sie die Aktivitäten und die Art der Beziehungen nach folgendem Anwendungsfalldiagramm:

****

1. Anwendungsfalldiagramm ergänzen

Ergänzen Sie die fehlenden Kanten im folgenden Anwendungsfalldiagramm. Verwenden Sie dabei <<include>> und <<extend>>-Abhängigkeiten sowie Vererbungsbeziehungen. Tipp: Sie können jeden Kantentyp mindestens einmal verwenden.



1. Anwendungsfälle aus Lastenheftauszug

Identifizieren Sie aus dem folgenden Lastenheftauszug die Akteure und Anwendungsfälle.

„Für eine Stadtbibliothek soll ein Softwaresystem entwickelt werden. Das Ausleihpersonal soll damit Bücher ausleihen, zurücknehmen und vorbestellen können. Weiterhin sollen Leser registriert werden. Die Bibliotheksverwaltung soll neue Bücher registrieren können und säumige Leser mahnen können.“

Identifizierte Akteure:

– Ausleihpersonal

– Bibliotheksverwaltung  
  
Identifizierte Anwendungsfälle:

– Bücher ausleihen

– Bücher zurücknehmen

– Bücher vorbestellen

– Leser registrieren

– Bücher registrieren

– Leser mahnen

1. Modellierung eines Use-Case-Diagramms

Modellieren Sie für ein Online-Auktions-System als Anwendungsfalldiagramm die folgend geschilderten Sachverhalte nach der Unified Modelling Language (UML):

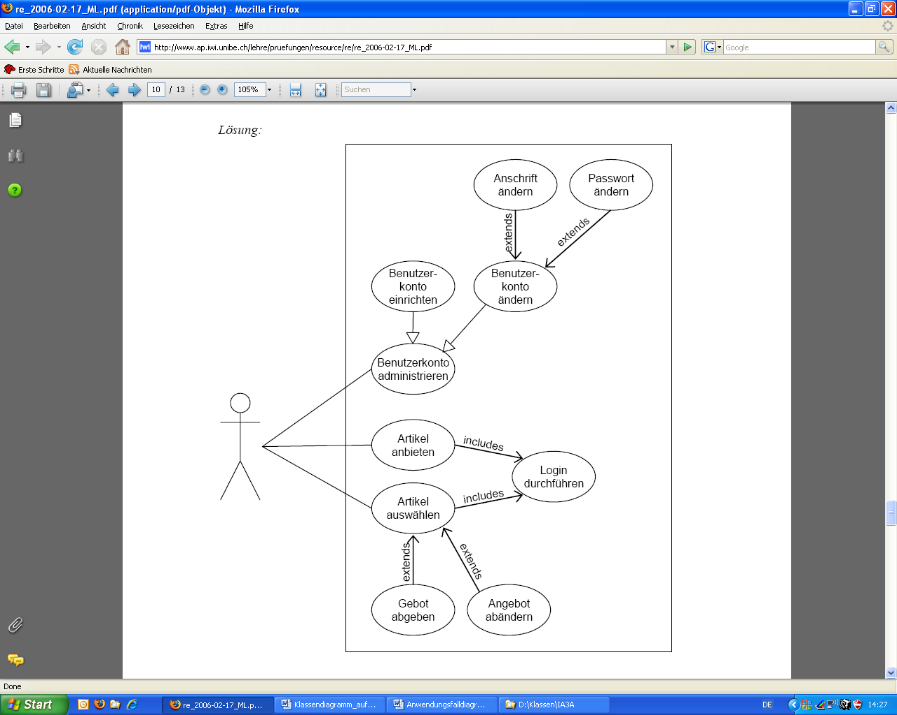
* Bevor ein Kunde das System benutzen kann, muss dieser sich zunächst ein neues/eigenes Benutzerkonto einrichten.
* Für die Nutzung jeder anderen Funktion des Auktions-Systems müssen sich die Benutzer zuvor anmelden.

Die weiteren Funktionen des Systems sollen die Benutzerkontoadministration, das Anbieten eines Artikels und das Bieten für einen Artikel sein:

* Im Rahmen der Benutzerkontoadministration werden die Benutzerdaten des jeweiligen Kunden angezeigt. Bei Bedarf ist es möglich, Anschrift und/oder Passwort zu ändern.
* Um einen Artikel anbieten zu können, müssen zunächst dessen Details eingegeben werden. Anschliessend kann das Angebot veröffentlicht werden.
* Veröffentlichte Angebote können abgeändert werden, solange noch keine Gebote vorliegen. Das System bietet diese Option an, wenn ein Benutzer einen Artikel ausgewählt hat, den er zu einem früheren Zeitpunkt selbst als Angebot erfasst hat.
* Um ein Gebot abgeben zu können, muss natürlich zunächst ein Artikelausgewählt werden.

Hinweise:

* Machen Sie – sofern sinnvoll – von einer Mehrfachverwendung einzelner Anwendungsfälle Gebrauch.
* Bitte begründen Sie unterhalb Ihres Anwendungsfalldiagramms stichwortartig die Wahl der einzelnen Assoziationen (extend, include oder Generalisierung/ Spezialisierung).
* Falls einzelne der genannten Teilfunktionen nicht explizit in Ihrem Anwendungsfalldiagramm ersichtlich sind (das ist im Einzelfall denkbar), so ist diese Designentscheidung zu begründen.



1. Vererbung

Wie kann folgender Sachverhalt mittels UML2 Standardnotation abgebildet werden:   
**Mann und Frau gehen im Garten Eden spazieren.**

Falls das Use-Case Diagramm diesen Sachverhalt nicht abbildet (🗵) begründen Sie Ihre Entscheidung durch Nennung der Differenzen zwischen Use-Case-Diagramm und textuellem Anwendungsfall.

|  |  |  |
| --- | --- | --- |
| 🞏 🗸  🞏 🗴 | ud_clipping13e.png | 🗴 *spazieren gehen wird von Mann oder Frau durchgeführt.* |
| 🞏 🗸  🞏 🗴 | ud_clipping13d.png | 🗴 Hier geht Frau gemeinsam mit Mann oder Frau spazieren (weil Frau eine Spezialisierung von Mann ist). |
| 🞏 🗸  🞏 🗴 | ud_clipping13b.png | 🗴 spazieren gehen wird von Mann oder Frau (oder einem Akteur namens Beide) durchgeführt. |
| 🞏 🗸  🞏 🗴 | ud_clipping13a.png | 🗸 |
| 🞏 🗸  🞏 🗴 | ud_clipping13c.png | 🗴 spazieren gehen wird von zwei Akteuren Mann, Frau bzw. Beide durchgeführt. Es kann also auch vorkommen, dass Frau mit Frau spazieren geht oder Mann mit Beide. |

*Wie kann folgender Sachverhalt mittels UML2 Standardnotation abgebildet werden:*   
**Eine Reparatur kann von einem Lehrling, einem Meister, oder einem anderen Mitarbeiter der Werkstatt durchgeführt werden.**

Falls das Use-Case Diagramm diesen Sachverhalt nicht abbildet (🗵) begründen Sie Ihre Entscheidung durch Nennung der Differenzen zwischen Use-Case-Diagramm und textuellem Anwendungsfall.

|  |  |  |
| --- | --- | --- |
| 🞏 🗸  🞏 🗴 | ud_clipping06b.png | 🗴Hier kann Reparatur durchführen nur von Meister oder Lehrling, nicht aber von einem anderen Mitarbeiter durchgeführt werden. |
| 🞏 🗸  🞏 🗴 | ud_clipping06a.png | 🗸 |
| 🞏 🗸  🞏 🗴 | ud_clipping06d.png | 🗴 Hier wird die Reparatur vom Lehrling gemeinsam mit dem Meister oder Lehrling durchgeführt(weil Lehrling eine Spezialisierung von Meister ist). |
| 🞏 🗸  🞏 🗴 | ud_clipping06c.png | 🗴 Hier sind Meister und Lehrling gemeinsam an Reparatur durchführen beteiligt. |
| 🞏 🗸  🞏 🗴 | ud_clipping06e.png | 🗴 Hier sind Meister, Lehrling und anderer MA gemeinsam an Reparatur durchführen beteiligt. |

*Wie kann folgender Sachverhalt mittels UML2 Standardnotation abgebildet werden:*   
**Ein Assistent führt mit einem Studenten ein Abgabegespräch durch. Im Zuge des Abgabegesprächs muss der Assistent immer eine Benotung durchführen.**

Begründen Sie bei den nicht zutreffenden Diagrammen, warum sie der Anwendungsfall-Beschreibung nicht entsprechen.

|  |  |  |
| --- | --- | --- |
| 🞏 🗸  🞏 🗴 | ud_clipping04c.png | 🗴 Hier erweitert Gespräch durchführen den Anwendungsfall Benotung durchführen, es ist also optional und somit nicht verpflichtend, dass im Zuge von Benotung durchführen der Anwendungsfall Gespräch durchführen ausgeführt wird. |
| 🞏 🗸  🞏 🗴 | ud_clipping04d.png | 🗴 Hier erbt Benotung durchführen von Gespräch durchführen. Benotung durchführen ist also eine Spezialisierung von Gespräch durchführen. |
| 🞏 🗸  🞏 🗴 | ud_clipping04b.png | 🗴 Der Pfeil von Gespräch durchführen zu Benotung durchführen entspricht nicht UML Notation. Stattdessen sollte eine include-Beziehung modelliert werden. |
| 🞏 🗸  🞏 🗴 | ud_clipping04a.png |  |

Wie kann folgender Sachverhalt mittels UML2 Standardnotation abgebildet werden:

**Eine Person geht essen. Im Zuge dessen könnte es notwendig sein, dass die Person Geld abheben muss.**

Falls das Use-Case Diagramm diesen Sachverhalt nicht abbildet (🗵) begründen Sie Ihre Entscheidung durch Nennung der Differenzen zwischen Use-Case-Diagramm und textuellem Anwendungsfall.

|  |  |  |
| --- | --- | --- |
| 🞏 🗸  🞏 🗴 | ud_clipping05f.png | 🗴Essen gehen erbt das Verhalten von Geld abheben und kann dieses überschreiben bzw. ergänzen. |
| 🞏 🗸  🞏 🗴 | ud_clipping05b.png | Hier ist die Pfeilrichtung falsch. Geld abheben sollte Essen gehen erweitern und nicht umgekehrt. |
| 🞏 🗸  🞏 🗴 | ud_clipping05a.png |  |
| 🞏 🗸  🞏 🗴 | ud_clipping05c.png | In diesem Fall muss die Person immer Geld abheben, wenn Essen gehen ausgeführt wird. |
| 🞏 🗸  🞏 🗴 | ud_clipping05e.png | Geld abheben erbt das Verhalten von Essen gehen und kann dieses überschreiben bzw. ergänzen. |
| 🞏 🗸  🞏 🗴 | ud_clipping05d.png | In diesem Fall muss die Person immer Essen gehen, wenn Geld abheben ausgeführt wird. |

Der folgende Text beschreibt, wie man aus einer Anforderungsbeschreibung systematisch ein objekt-orientiertes Analysemodell in Form von UML-Diagrammen erstellen kann. Fügen Sie inhaltlich passende Wörter in die durch einen Strich gekennzeichneten Lücken ein. Pro Lücke fehlt genau ein Wort. Unpassende Wörter führen zu Punktabzügen.

Finden Sie zunächst Akteure, d.h. alles, was mit dem System interagiert, aber selbst nicht Teil des \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Systems ist. Identifizieren Sie danach für jeden \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Akteur, was er mit dem System tun will und bilden Sie daraus einen Anwendungsfall. Jede genannte Anforderung muss damit abgedeckt sein. Schreiben Sie für jeden \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Anwendungsfall ein Szenario bestehend aus einer Vorbedingung, einer nummerierten Abfolge der Tätigkeiten und einer \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Nachbedingung. Ein solches Szenario können Sie auch grafisch in Form eines \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Sequenzdiagramms darstellen. Daraus ergehen die Objekte, die interagieren. Definieren Sie durch Verallgemeinerung zu den so ermittelten Objekten passende \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Klassen. Fügen Sie dies grafisch zu einem Klassendiagramm zusammen. Modellieren Sie auch \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Assoziationen zwischen den Klassen. Ergänzen Sie zusätzlich die \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Operationen, die sich aus dem Verhalten in den Szenarien ergeben. Wenn möglich, beschreiben Sie danach jeweils grafisch in einem \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Zustandsdiagramm den dazugehörigen Lebenszyklus.

1. Multiple Choice...

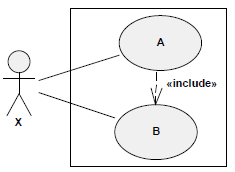
**a) Welche der folgenden Aussagen charakterisieren Anwendungsfälle? (\_\_\_\_P / 2 P)**

|  |  |
| --- | --- |
| 🞏 🗸🗸 / 🞏 🗴 | Anwendungsfälle ergeben sich aus den Wünschen der Kunden. |
| 🞏 🗸 / 🞏 🗴🗴 | Anwendungsfalldiagramme sind dazu geeignet, die Schnittstellen zwischen zwei verschiedenen Systemen zu modellieren. |
| 🞏 🗸 / 🞏 🗴🗴 | Anwendungsfälle beschreiben einen prozeduralen Ablauf in einem System. |
| 🞏 🗸🗸 / 🞏 🗴 | Anwendungsfälle beschreiben die Funktionalität bzw. das Verhalten, welches von dem zu entwickelnden System erwartet wird. |

**b) Welche Anwendungsfälle kann Akteur T (egal ob allein oder gemeinsam mit anderen Akteuren) ausführen? (\_\_\_\_P / 2 P)**

|  |  |
| --- | --- |
| https://www.big.tuwien.ac.at/oom/UCD/akteure1.png | 🞏 🗸 / 🞏 🗴🗴 Anwendungsfall D  🞏 🗸🗸 / 🞏 🗴 Anwendungsfall E  🞏 🗸🗸 / 🞏 🗴 Anwendungsfall F |

**c) Folgende <<include>>-Beziehung bedeutet, dass (\_\_\_\_P / 4 P)**

🞏 🗸 / 🞏 🗴🗴 ... das Verhalten von A in B eingefügt werden kann aber nicht muss.

🞏 🗸 / 🞏 🗴🗴 ... B an Stelle von A ausgeführt wird.

🞏 🗸 / 🞏 🗴🗴 ... B nicht ohne A ausgeführt werden kann.

🞏 🗸 / 🞏 🗴🗴 ... A den Anwendungsfall B aufrufen kann, aber nicht muss.

🞏 🗸 / 🞏 🗴🗴 ... das Verhalten von B in A eingefügt werden kann aber nicht muss.

🞏 🗸🗸 / 🞏 🗴 ... A nicht ohne B ausgeführt werden kann.

🞏 🗸🗸 / 🞏 🗴 ... das Verhalten von B immer in A eingefügt werden muss.

🞏 🗸 / 🞏 🗴🗴 ... das Verhalten von A immer in B eingefügt werden muss.

**d) Akteure in einem Anwendungsfalldiagramm ...**

🞏 🗸🗸 / 🞏 🗴 ... können durch Assoziationen mit abstrakten und nicht-abstrakten Anwendungsfällen in Beziehung gesetzt werden.

🞏 🗸🗸 / 🞏 🗴... interagieren mit dem beschriebenen System.

🞏 🗸🗸 / 🞏 🗴... können durch Vererbung miteinander in Beziehung stehen.

🞏 🗸🗸 / 🞏 🗴... können das beschriebene System benutzen.

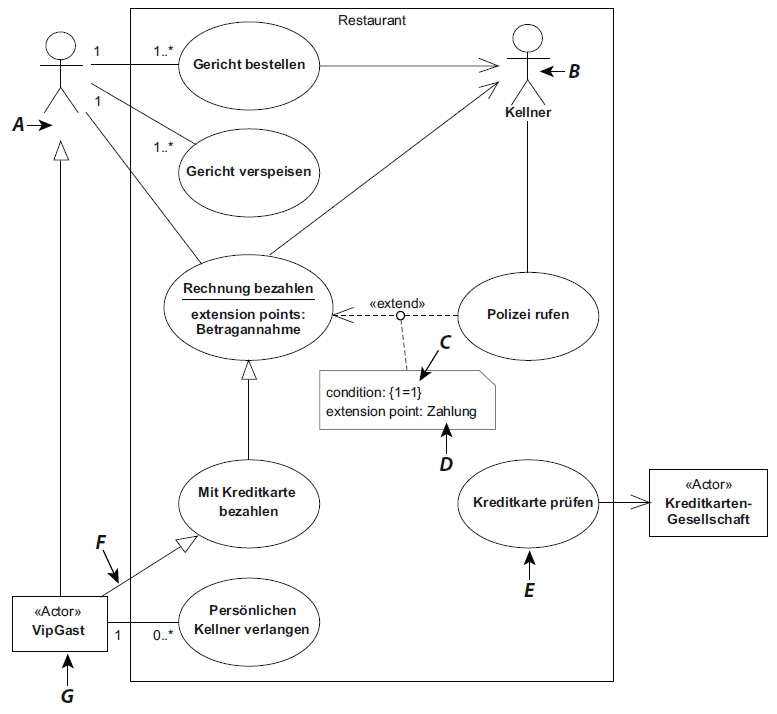
🞏 🗸🗸 / 🞏 🗴... können vom beschriebenen System benutzt werden.

🞏 🗸 / 🞏 🗴🗴... können mit Anwendungsfällen und untereinander kommunizieren.

🞏 🗸 / 🞏 🗴🗴... befinden sich immer innerhalb des beschriebenen Systems.

🞏 🗸 / 🞏 🗴🗴.... interagieren mit Hilfe von <<include>>-Beziehungen mit dem System.

1. Fehleranalyse



**A: Name des Akteurs fehlt:** Akteure müssen einen Namen erhalten, der ihre Rollen eindeutig definiert.

**B: Akteur innerhalb des Systems:** Akteure definieren Rollen, die gegenüber einem System auftreten, und können damit nicht als ein Teil des Systems modelliert werden.

**C: Falsche Bedingung oder falsche Erweiterungsbeziehung:** Die Bedingung in der Anmerkung ist immer wahr (true), weshalb die Erweiterung immer durchgeführt wird. Soll der Anwendungsfall Gericht bezahlen nur in speziellen Fällen erweitert werden, muss die Bedingung korrigiert werden. Ist es gewünscht, die Funktionalität des Anwendungsfalls Polizei rufen immer einzubinden, ist eine <<include>>-Beziehung die bessere Wahl.

**D: Unbekannter Erweiterungspunkt:** Die Anmerkung weist dem erweiternden Anwendungsfall Polizei rufen einen Erweiterungspunkt Zahlung zu, der im erweiterten Anwendungsfall Gericht bezahlen nicht definiert ist. Einem erweiternden Anwendungsfall dürfen nur bereits definierte Erweiterungspunkte zugewiesen werden.

**E: Nicht durchführbarer Anwendungsfall:** Der Anwendungsfall Kreditkarte prüfen besitzt eine gerichtete Assoziation zu dem Akteur Kreditkarten-Gesellschaft, weshalb diese den Anwendungsfall selbst nicht initiieren kann. Die Funktionalität des Anwendungsfalls kann nicht abgerufen werden, weil er weder weitere Assoziationen zu Akteuren noch Beziehungen zu Anwendungsfällen hat. Überprüfen Sie in solchen Fällen das Diagramm daraufhin, ob Assoziationen fehlen oder der Anwendungsfall entfernt werden kann.

**F: Falsche Generalisierung** Generalisierungen dürfen nur zwischen Akteuren oder zwischen Anwendungsfällen modelliert werden, nicht zwischen Akteuren und Anwendungsfällen.

Soll eine Kommunikationsrichtung modelliert werden, sollte die gerichtete Navigation verwendet werden.

**G: Rechtecksymbol für einen menschliche Akteur:** Es empfiehlt sich, für menschliche Akteure das »Strichmännchensymbol« und für externe Systeme das »Rechtecksymbol« zu modellieren. Die UML schreibt eine solche Verwendung zwar nicht vor, sie hilft Ihnen jedoch, die Art der Akteure zu unterscheiden, und macht das Diagramm damit leichter verständlich.