

<input checked="" type="checkbox"/> DES3UEG1: Glock	Name <u>Elias Leonhardsberger</u>	Aufwand in h <u>5</u>
<input type="checkbox"/> DES3UEG3: Werth	Punkte _____	Kurzzeichen Tutor*in _____

---

**Hinweise und Richtlinien:**

- Übungsausarbeitungen müssen den im e-Learning angegebenen Formatierungsrichtlinien entsprechen – Nichtbeachtung dieser Formatierungsrichtlinien führt zu Punkteabzug.
- Treffen Sie, falls notwendig, sinnvolle Annahmen und dokumentieren Sie diese nachvollziehbar in ihrer Lösung.
- Für die Beispiele empfehlen wir Ihnen den Einsatz von Bleistift und Papier. Es steht Ihnen jedoch frei, ob Sie ein Softwareprogramm verwenden.
- Bitte beachten Sie zusätzlich zu den allgemeinen Formatierungsrichtlinien:
  - Vergessen Sie nicht Multiplizitäten, Primärschlüssel und Generalisierungsbeziehungen anzugeben!
  - Beschriften Sie Assoziationen!
  - Achten Sie auf eine einheitliche Benennung und Schreibweise der Klassen, Attribute und Assoziationen, entscheiden Sie sich für entweder durchgängig Englisch oder Deutsch.
  - Bei Gruppenarbeit geben Sie die Namen der Gruppenmitglieder an.

---

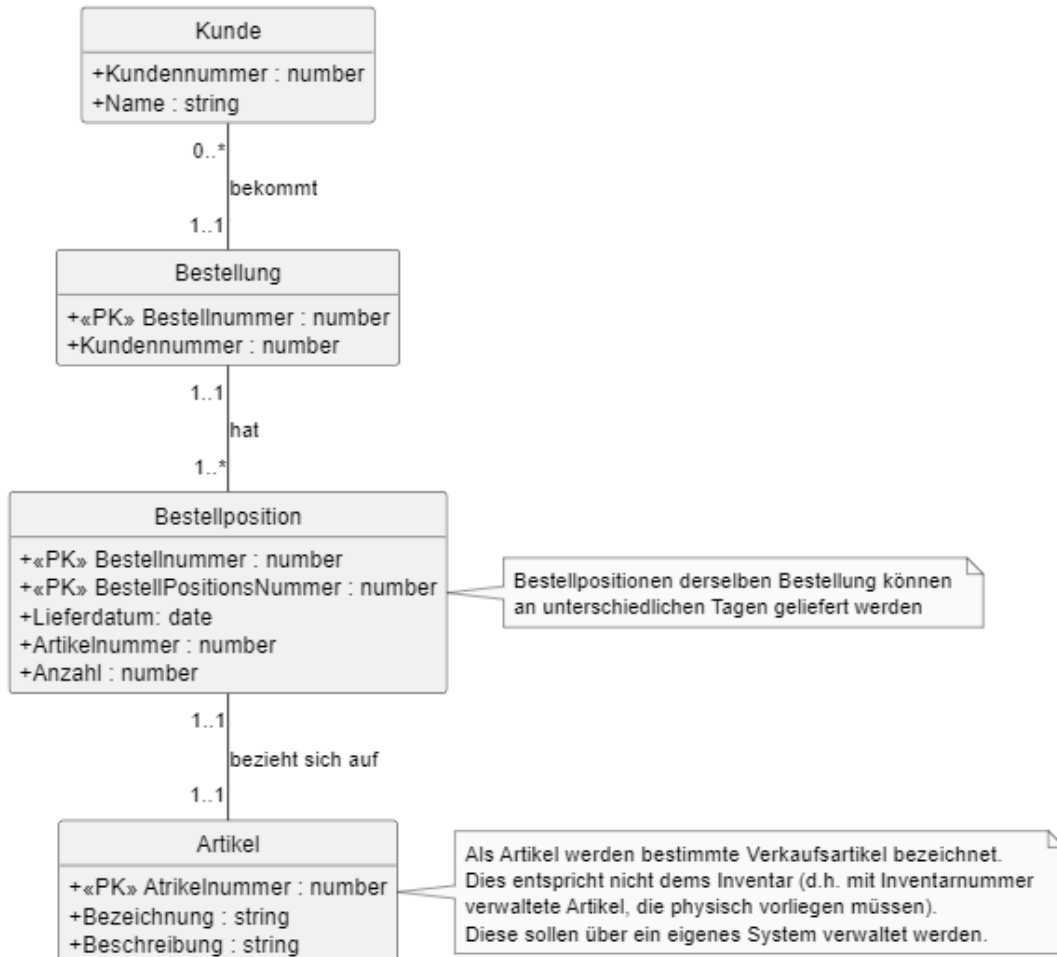
**Ziel dieser Übung ist es, aus einer textuellen Beschreibung ein konzeptuelles Datenbankschema mit UML zu erstellen.**

## 1. UML Korrekturen

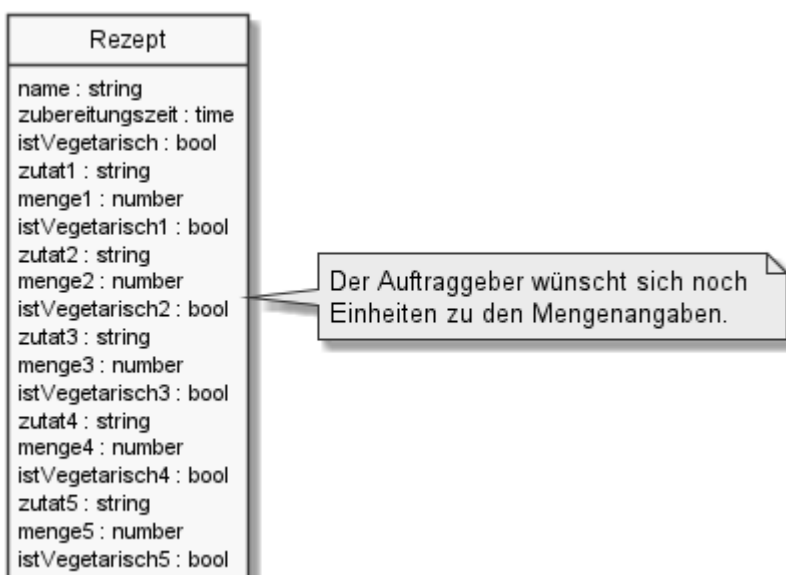
(je 2 Punkte, 8 Punkte)

Überprüfen Sie folgende UML-Klassendiagramme und beheben Sie die Fehler. Geben Sie die korrigierten UML Klassendiagramme ab und geben Sie zusätzlich in der Lösung an, welche Änderungen Sie warum durchgeführt haben.

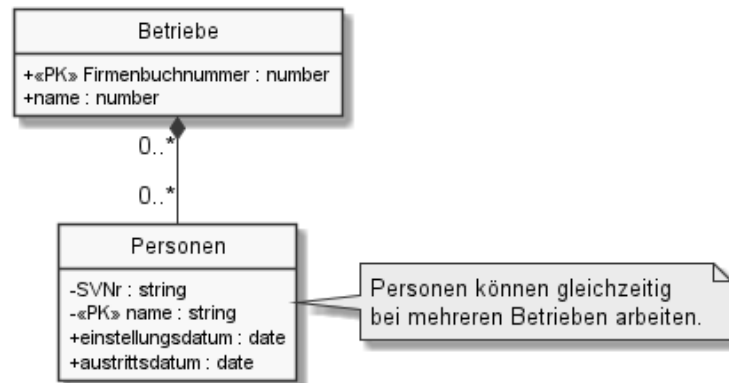
### 1.1 Bestellungen



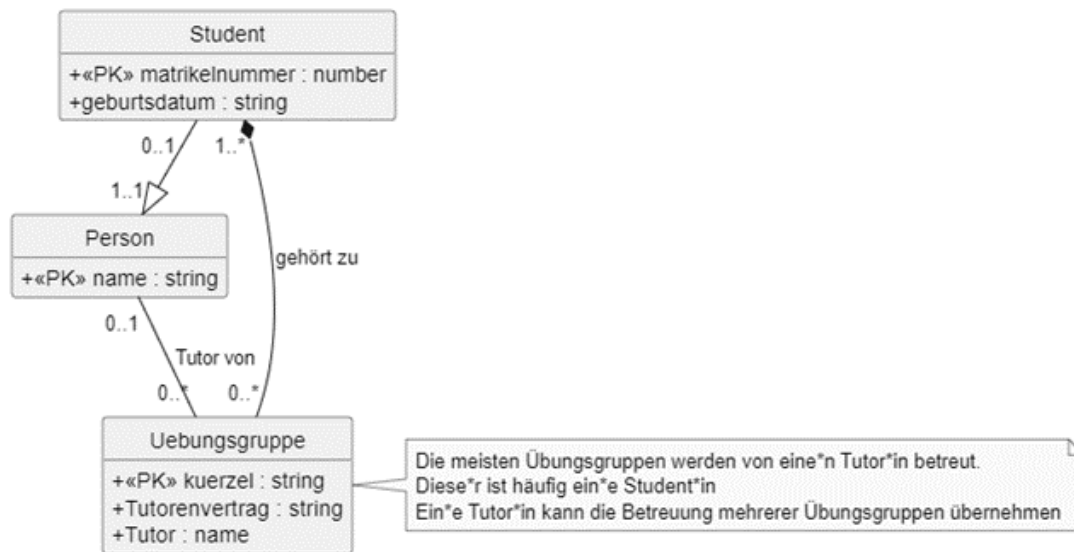
### 1.2 Rezeptverwaltung



## 1.3 Anstellungsverhältnisse



## 1.4 Studierende

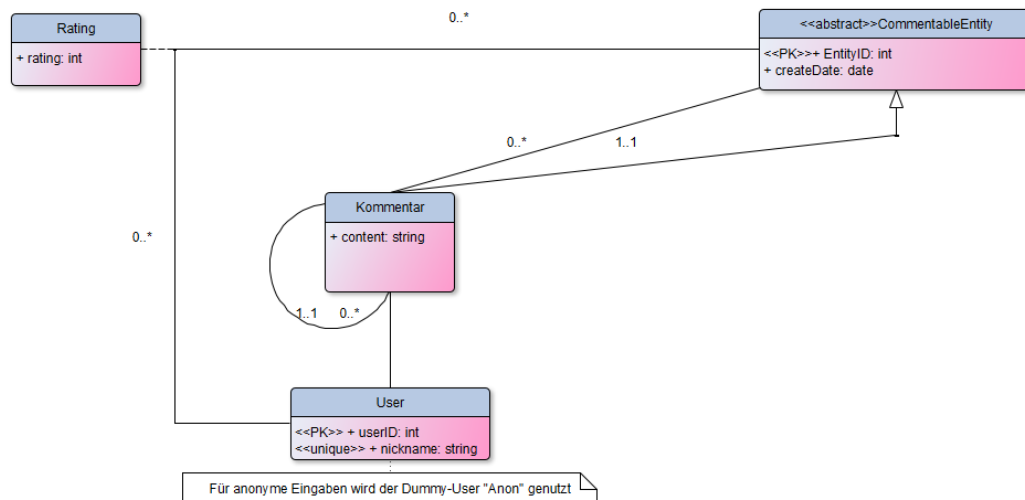


## 2. Studentisches Hilffssystem

(8 Punkte)

Helfen Sie den Studenten der nächsten Jahrgänge, indem Sie ein System entwickeln, in dem relevante Informationen über das Studium gespeichert werden. Relevant sind vor allem, welche Angaben (z.B.: Hausübungen, Kurzttests, Klausuren) in den letzten Jahren gegeben wurden wie diese zu lösen waren und wie benote wurde. Außerdem sollen die jeweiligen LVAs und Lehrenden mitverwaltet werden. Erstellen Sie in einem ersten Schritt ein **UML-Diagramm**.

- Wir verwenden „Studi-CMS“ das beim Einrichten bereits einige Tabellen erstellt. User erstellen Ratings (1-5) und Kommentare zu „CommentableEntities“, wobei Kommentare selbst CommentableEntities sind.



- Lösungen, Angaben, LVAs* und *Lehrende* sollen bewertbar- und kommentierbar sein.
- LVAs* repräsentieren Lehrveranstaltungen mit ihren generellen Eigenschaften (Titel, Beschreibung), während die einzelnen *Abhaltungen* LVAs pro Semester repräsentieren. Kodieren Sie das Semester als String („WS2021“, „SS2024“, ...).
- Zu jeder *Abhaltung* soll erfasst werden welche *Lehrenden* dafür verantwortlich sind. Es können auch mehrere Lehrende an derselben Abhaltung mitarbeiten. Speichern Sie für jeden Lehrenden den Namen und das Department als String.
- Angaben* werden zur jeweiligen *Abhaltung* zugeordnet und enthalten einen Dateipfad, das Datum, wann die Angabe ausgegeben wurde, und ob es sich um eine Vorlesungsklausur handelt.
- User* können zu jeder *Angabe*, ihre *Note(n)* hinterlegen und/oder *Lösungen* zur Angabe hochladen (Speichern Sie zu jeder Lösung sowohl ein Textfeld als auch einen Dateipfad). Achten Sie darauf, dass auch User Lösungen erstellen können, die keine Note erhalten haben und vice versa. Wenn Sie einen User oder eine Angabe löschen sollen alle damit verbundenen Lösungen und Noten ebenfalls

Stellen Sie sicher, dass folgende Test-Abfragen in Ihrem Modell möglich sind und geben Sie dafür **SQL-Statements** an (wählen Sie geeignete Namen für Beziehungs-Tabellen):

- Das mittlere Rating vom Lehrenden "Bernhard Werth"
- Alle Angaben der LVA "DES3UE" vom Semester "WS2023"
- Die mittlere Note aller Aufgaben zur LVA mit dem Titel "AMS4UE" pro Jahr
- Alle Lösungen zur LVA mit dem Titel "DES3VL" sortiert nach Jahr
- Die Anzahl aller anonymen Noten für die Angabe mit der EntityID 14

### 3. Mensa-Datenbankmodell

(8 Punkte)

Eine Mensa will zukünftig eine Datenbank für den laufenden Betrieb einsetzen. Ergänzen Sie basierend auf der nachfolgenden Beschreibung das gegebene UML-Klassendiagramm für den Datenbankentwurf. Achten Sie dabei auf die Besonderheiten beim Datenbankentwurf mit UML.

Die Mensa hat verschiedene Speisen und Getränke im Angebot. Zunächst einmal soll gespeichert werden, welche Speisen verkauft werden. Eine Speise hat einen Namen und eine Portionsgröße (klein, normal, groß), die in Kombination eindeutig sind, weiters wird der Preis gespeichert.

Auch bei den Getränken setzt sich der Schlüssel aus Namen und Größe zusammen (z.B. Cola; 0,5l), zusätzlich sind der Preis und die vorrätige Menge bekannt.

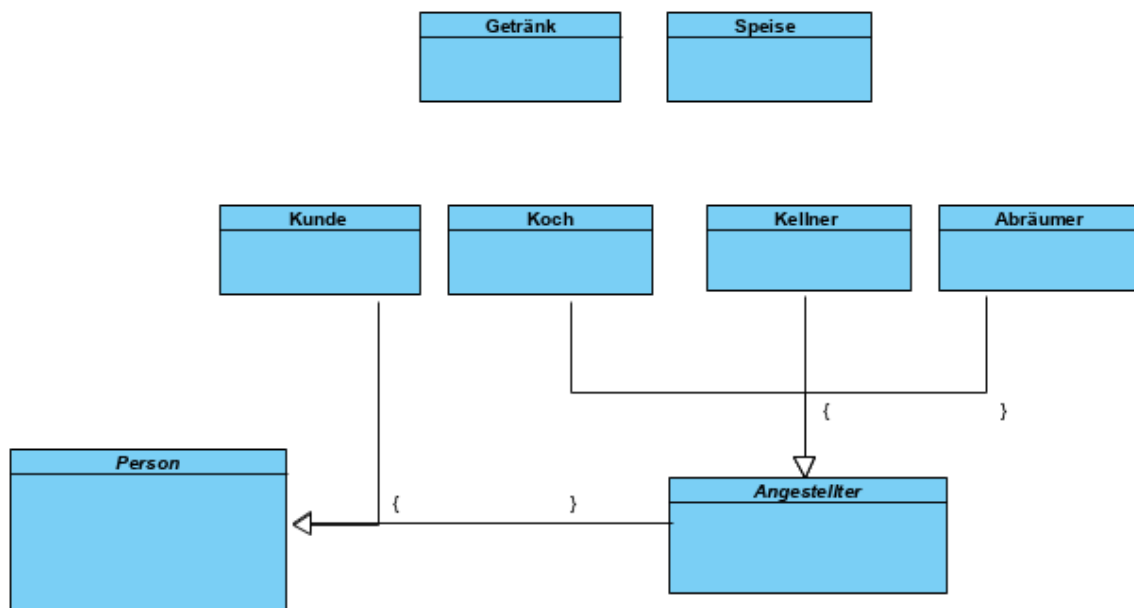
Eine Speise besteht aus verschiedenen Zutaten, wobei jede Speise mindestens drei Zutaten hat. Zu jeder Speise wird in einem Rezept vermerkt, wieviel von den jeweiligen Zutaten für die Zubereitung benötigt wird. Eine Zutat hat einen eindeutigen Namen und es ist der vorhandene Vorrat bekannt.

Auch die Angestellten sollen erfasst werden. Jede\*r Angestellte hat Vor- und Nachname, eine eindeutige Sozialversicherungsnummer, eine Adresse und es wird das Gehalt des Angestellten vermerkt. Die Angestellten werden in Gruppen eingeteilt: Kellner, Abräumer und Köche. Bei den Köchen wird die Qualifikation (die Anzahl der Hauben) in der Datenbank erfasst.

In der Mensa gibt es viele Tische. Jeder Tisch hat eine eindeutige Nummer, es wird zusätzlich die Anzahl der maximalen Sitzplätze gespeichert. Jede\*r Abräumer\*in wird für fünf bis maximal zehn Tische eingeteilt, Tische werden immer von mindestens einem/r Abräumer\*in betreut.

Die Gäste wählen aus dem Angebot beliebig viele Speisen und Getränke, wobei ihre Bestellung in die Datenbank eingetragen wird: Jede Bestellung wird dem Tisch zugeordnet, an dem sie getätigt wurde. Die Tischnummer und ein Zeitstempel bestimmen zusammen jede Bestellung eindeutig. Zusätzlich wird der Betrag notiert, der von den Gästen zu begleichen ist. Ob ein Tisch diesen Betrag bereits gezahlt hat, wird ebenso bei der Bestellung vermerkt. Darüber hinaus wird gespeichert, welche Kellner die Bestellung bearbeiten, wobei jeder Bestellung mindestens ein Kellner zugeteilt ist.

Jedem/r Kunden/in wird mittels Studentenausweises eine eindeutige ID zugeordnet. Darüber hinaus werden Vorname, Nachname, Telefonnummer und Adresse vermerkt. Schlussendlich wird noch für alle Angestellte gespeichert, von welchen Angestellten sie bei Erkrankung oder Inanspruchnahme von Urlaubszeit vertreten werden, wobei für jeden Angestellten mindestens zwei Personen als Vertreter vermerkt sind und jeder Angestellte beliebig viele Angestellte vertreten kann.



# DES3UE Übung 1

Elias Leonhardsberger

7. Oktober 2024, Hagenberg

## Inhaltsverzeichnis

<b>1</b>	<b>UML Korrekturen</b>	<b>7</b>
1.1	Bestellungen . . . . .	7
1.2	Rezeptverwaltung . . . . .	8
1.3	Anstellungsverhältnisse . . . . .	9
1.4	Studierende . . . . .	10
<b>2</b>	<b>Studentisches Hilffsystem</b>	<b>11</b>
2.1	Klassendiagramm . . . . .	11
2.2	SQL-Statements . . . . .	12
<b>3</b>	<b>Mensa-Datenbankmodell</b>	<b>13</b>

# 1 UML Korrekturen

## 1.1 Bestellungen

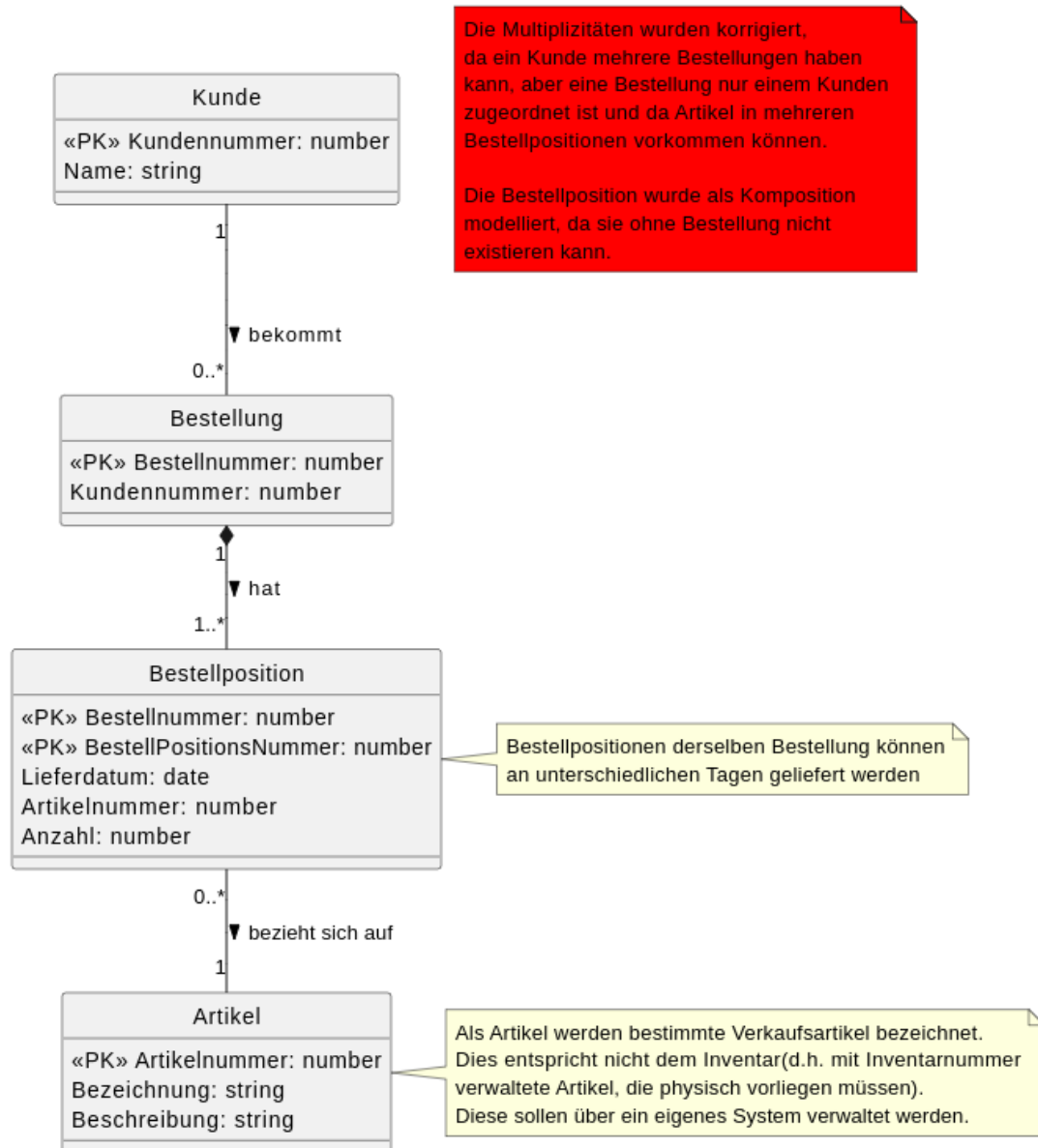


Abbildung 1: Verbessertes UML Diagram der Aufgabe 1.1 mit Begründungen

## 1.2 Rezeptverwaltung

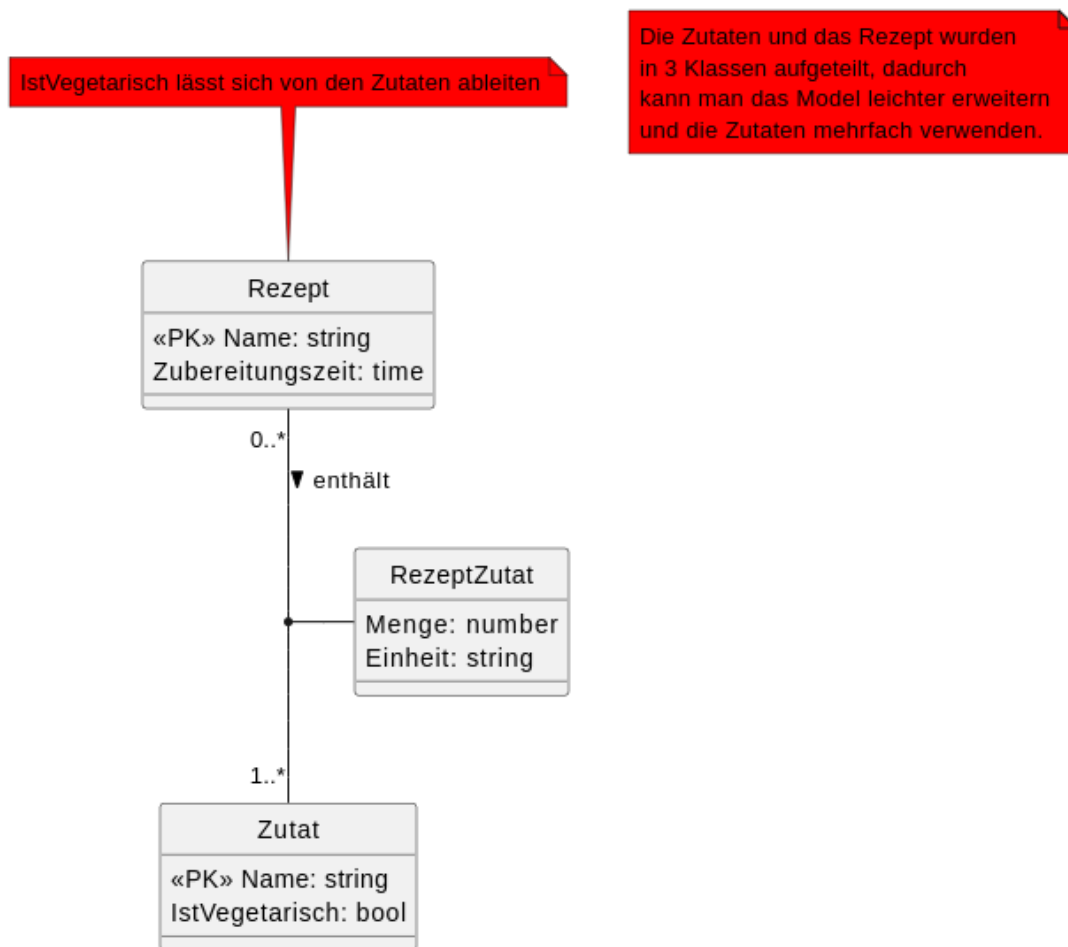
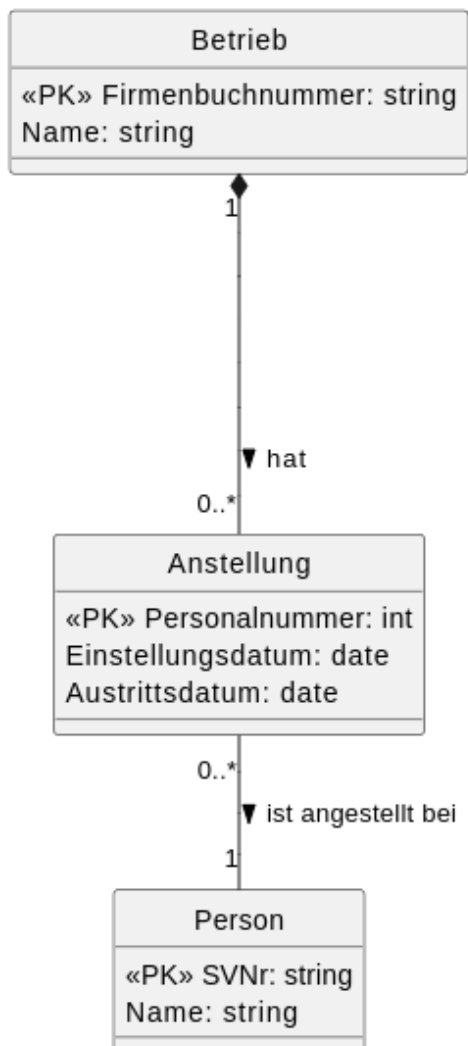


Abbildung 2: Verbessertes UML Diagram der Aufgabe 1.2 mit Begründungen



## 1.3 Anstellungsverhältnisse



Die Klassennamen wurden singularisiert und Datentypen von Attributen wurden angepasst.

Die Sozialversicherungsnummer wurde als Primärschlüssel gewählt, da sie eindeutig ist.

Annahme: Eine Person kann bei einem Betrieb austreten und wieder eintreten, dabei wird ein zweiter Eintrag mit Einstellungsdatum und Austrittsdatum angelegt.

Daher wurde eine Klasse zwischen Person und Betrieb eingefügt.

Abbildung 3: Verbessertes UML Diagram der Aufgabe 1.3 mit Begründungen

## 1.4 Studierende

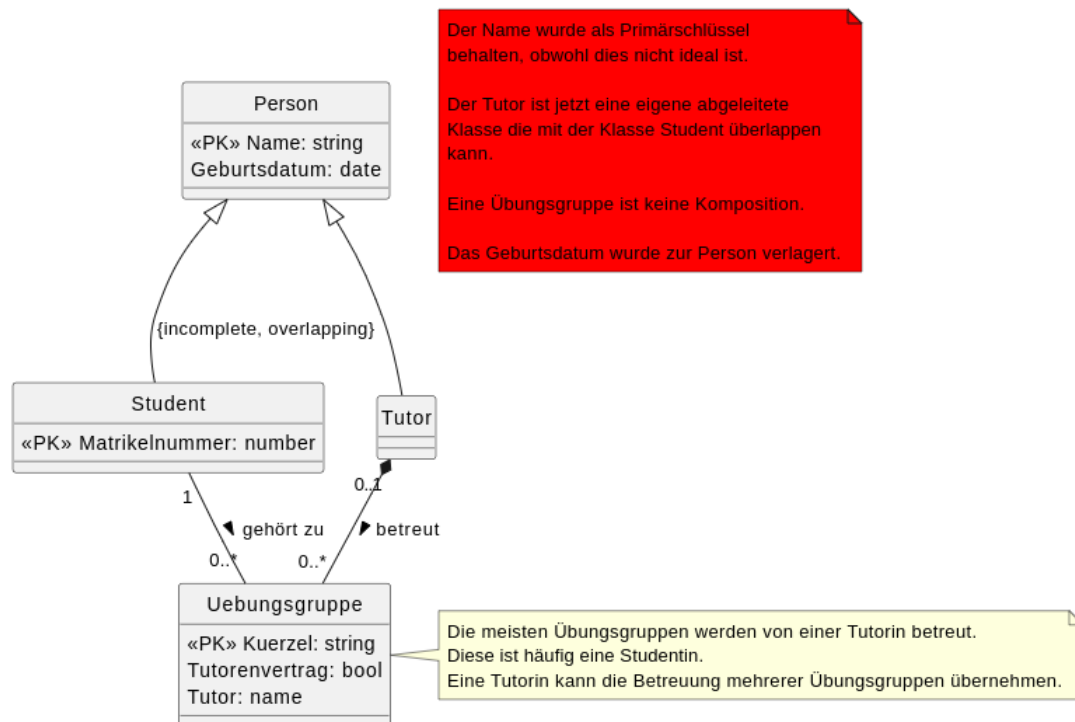


Abbildung 4: Verbessertes UML Diagram der Aufgabe 1.4 mit Begründungen

## 2 Studentisches Hilffssystem

### 2.1 Klassendiagramm

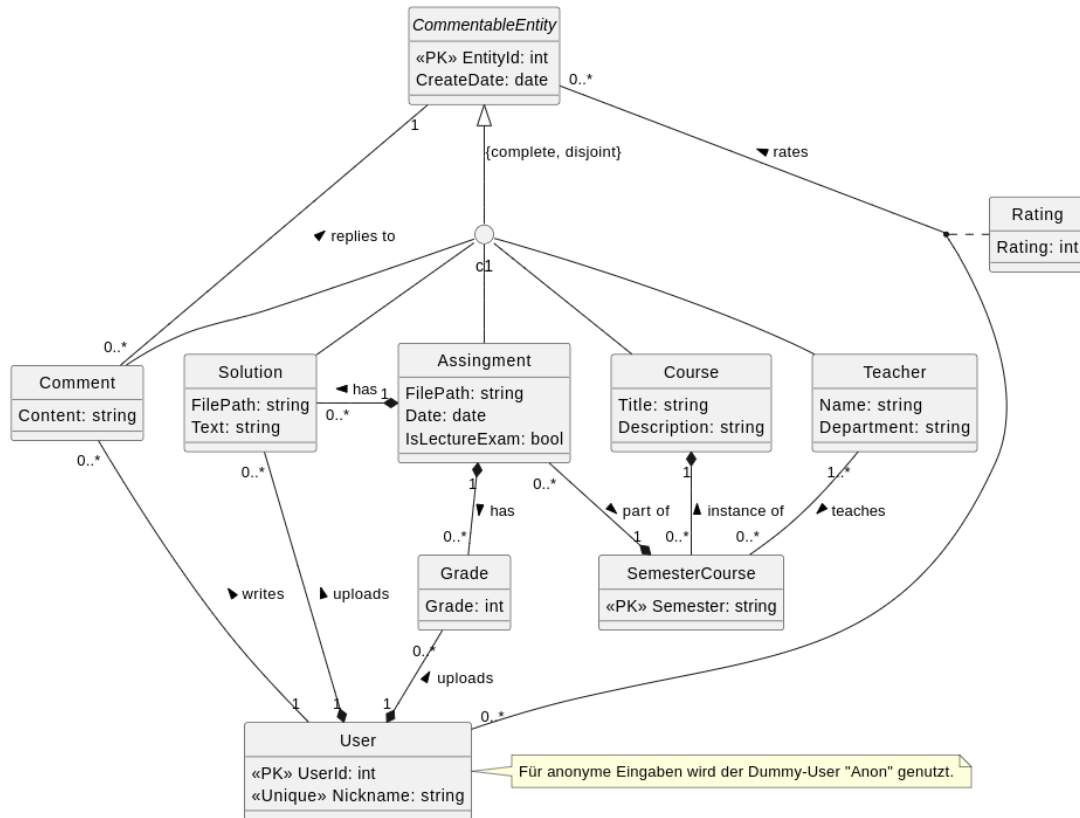


Abbildung 5: Klassendiagramm der Aufgabe 2

Das Basisdiagramm wurde ein wenig angepasst, da *Comments* bereichs kommentierbar sind, müsste es keine selbstreferenzierende Relation von *Comments* geben.

*Grades* und *Solutions* sind keine Beziehungsentitäten, da *User* mehrere *Grades* und *Solutions* pro *Assignment* hochladen können, dies ist vor allem für den „Anon“ *User* relevant.

Ein Kreis *c1* wurde verwendet um die Vererbung mit PlantUML schöner darzustellen.

Da die Sprache im Basisdiagramm gemischt war, wurde englisch als Sprache festgelegt entschieden.

## 2.2 SQL-Statements

```
1  -- 1
2  SELECT AVG(r.Rating) AS AverageRating
3  FROM Teacher t
4  INNER JOIN Rating r
5      ON r.EntityId = t.EntityId
6  WHERE t.Name = 'Bernhard Werth';
7
8  -- 2
9  -- SemesterCourse has a combined PK of CourseId(from Course EntityId)
   ↳ and Semester
10 SELECT a.FilePath
11      ,a.DATE
12      ,a.IsLectureExam
13 FROM ASSIGNMENT a
14 INNER JOIN Course C
15      ON a.CourseId = C.EntityId
16 WHERE a.Semester = "WS2023"
17      AND C.Name = "DSE3UE";
18
19 -- 3
20 SELECT AVG(G.Grade) AS AverageGrade
21 FROM Grade G
22 INNER JOIN ASSIGNMENT a
23      ON G.AssignmentId = a.EntityId
24 INNER JOIN Course C
25      ON a.CourseId = C.EntityId
26 WHERE C.Name = "AMS4UE";
27
28 -- 4
29 SELECT s.FilePath
30      ,s.TEXT
31 FROM Solution s
32 INNER JOIN ASSIGNMENT a
33      ON s.AssignmentId = a.EntityId
34 INNER JOIN Course C
35      ON a.CourseId = C.EntityId
36 WHERE C.Name = "DES3VL"
37 ORDER BY s.DATE DESC;
38
39 -- 5
40 SELECT COUNT(*)
41 FROM Grade G
42 INNER JOIN USER u
43      ON G.UserId = u.UserId
44 WHERE u.Nickname = 'Anon'
```

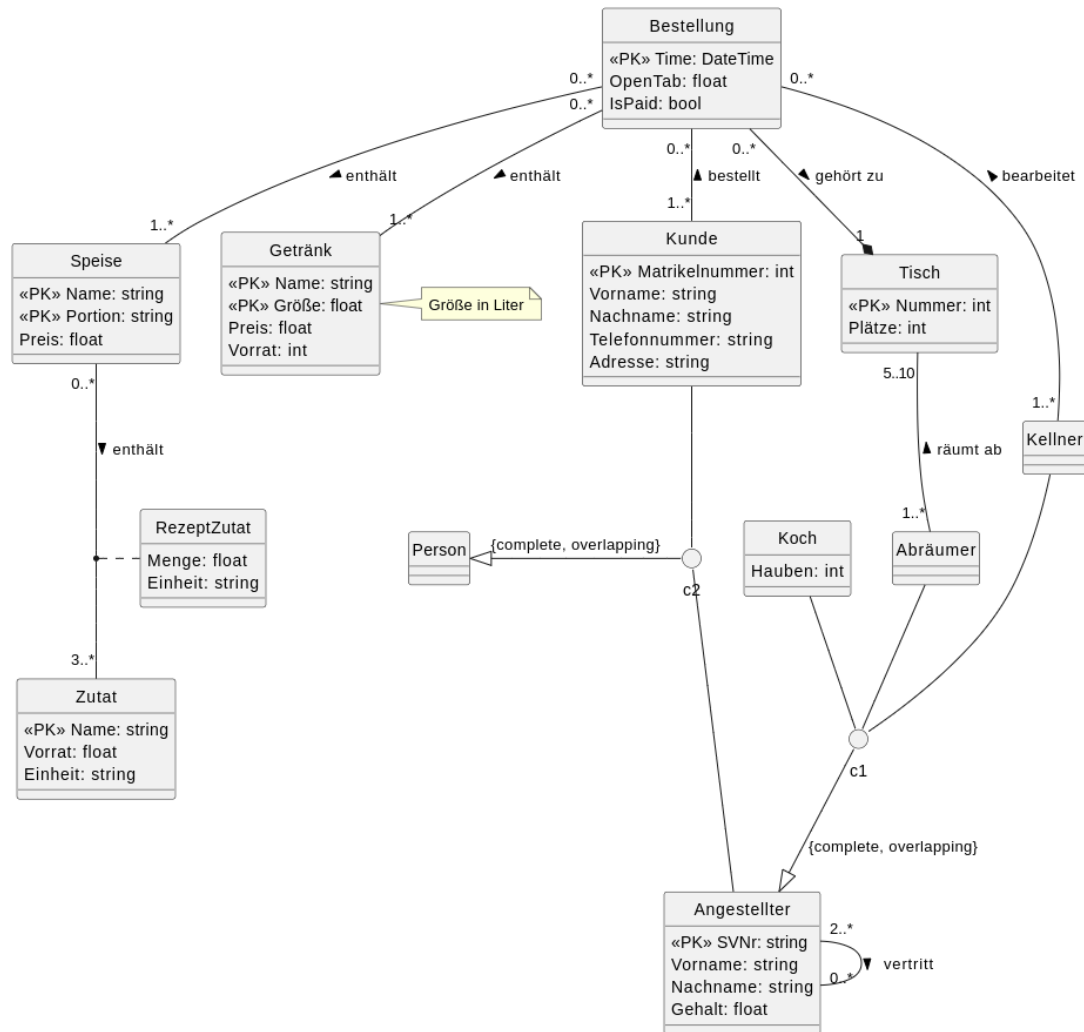


Abbildung 6: Klassendiagramm der Aufgabe 3

### 3 Mensa-Datenbankmodell

Es wurde angenommen, dass ein *Angestellter* auch als *Kunde* in der Mensa essen kann und, dass ein *Angestellter* mehrere Rollen(*Koch*, *Kellner*, *Abräumer*) haben kann.