

Planen eines Softwareproduktes

1. Aufgabe

- a) Den Befürchtungen des Verkaufspersonals kann durch folgende beispielhafte Schritte entgegengewirkt werden:
- Transparente Darstellung der Projektziele: Durch eine vollständig transparente und offene Kommunikation aller Projektbestandteile und -ziele soll von Anfang an offengelegt werden, was die Ziele des Projektes „BESUCHERAPP“ sind und welche Ziele explizit nicht vom Projekt verfolgt werden.

Frühzeitige Einbindung des Verkaufspersonals in das Projekt: Durch die ständige Mitarbeit und Miteinbeziehung des Verkaufspersonals kann verhindert werden, dass Gerüchte entstehen. Außerdem kann hierdurch das Verkaufspersonal in alle projektrelevanten Abstimmungen miteinbezogen werden, Maßnahmen die zum Abbau von Arbeitsplätzen führen, können so frühzeitig abgelehnt werden.

Zusicherungen gewähren: Durch verbindliche Zusicherungen und Verträge kann sichergestellt werden, dass keine Arbeitsplätze abgebaut werden. Es könnten neue Tätigkeiten in die Arbeitsverträge aufgenommen werden, um so auch langfristig festzuschreiben, dass das Verkaufspersonal weiterhin benötigt wird.

Weitere Lösungen sind möglich, die Beschreibung der Schritte kann kürzer ausfallen.

b)

| Risiko | Ursache | Auswirkung |
|--|--|---|
| Unterschätzung des Entwicklungsaufwandes | Keine Erfahrung bei der AMAG Soft GmbH | Projektende verzögert und das Projekt verteuert sich |
| Inkompatible Software-Schnittstellen | <ul style="list-style-type: none"> – Fehlerhafte bzw. lückenhafte Anforderungsanalyse – Unzureichende technische Vorklärungen | <ul style="list-style-type: none"> – Benötigte Funktionen können nicht eingebunden werden – Technische Machbarkeit der App kann nicht gewährleistet werden – Es entstehen Mehrkosten |
| Widerspruch des Personalrates | <ul style="list-style-type: none"> – Unzureichende Einbindung des Personalrates – Fehlende Transparenz der Projektziele – Personalrat fürchtet Arbeitsplatzkontrolle und den Verlust von Arbeitsplätzen | <ul style="list-style-type: none"> – Abbruch des Projektes – Erhöhter Abstimmungsbedarf und dadurch Verschiebung des Projektes |

Anhand der Aufgabenstellung geht nicht klar hervor, wie viele Auswirkungen und Ursachen zu nennen sind, daher ist davon auszugehen, dass jeweils eine Auswirkung / Ursache je Zelle ausreicht.

Weitere Lösungen sind möglich.

- c) Eine funktionale Anforderung wäre z. B., dass gewährleistet wird, dass die App auf allen gängigen mobilen Systemen (iOS sowie Android) lauffähig ist. Nur so können alle potenziellen Anwender angesprochen werden.

Eine beispielhafte nicht funktionale Anforderung wäre, dass die angebotenen Services und Funktionen der App jederzeit zur Verfügung stehen müssen. Andernfalls wäre nicht sichergestellt, dass z. B. jederzeit Eintrittskarten gekauft werden können.

Weitere mögliche funktionale Anforderungen:

- Berechnung der Einzelpreise und des Gesamtpreises für die Buchung
- Gewährleistung einer sicheren Authentifizierung
- Integration einer geeigneten Bezahlösung
- Push-Benachrichtigungen
- Individuelle Einstellungsmöglichkeiten zur App-Oberfläche

Weitere Lösungen sind möglich.

Weitere mögliche nichtfunktionale Anforderungen:

- Zuverlässige Benutzbarkeit
- Leichte Bedienung (Usability)
- Fehlertoleranz durch Auto-Vervollständigung und Feldprüfungen
- Gewährleistung der Datenintegrität durch Datenbankmechanismen

Weitere Lösungen sind möglich.

d) **Personalkosten:**

- Kosten für Mitarbeiter der AMAG Soft GmbH
- Kosten für das Kartenverkaufspersonal
- Kosten für Mitarbeiter der hauseigene IT-Abteilung
- Kosten für den Projektleiter der Parkanlage

Softwarekosten:

- Kosten für Entwicklungsumgebungen und Testinstanzen zur App
- Kosten für Softwarelizenzen

Hardwarekosten:

- Lesegeräte für die digitalen Tickets
- Serverkosten (Mailserver, Applikationsserver etc.)

Kommunikationskosten:

- Leitungs- und Übertragungskosten

Qualifizierungskosten:

- Schulungskosten für Mitarbeiterqualifizierungen (Kartenverkaufspersonal, Mitarbeiter der IT-Abteilung)
- Kosten für Trainer bei In-House-Schulungen
- Zertifikatskosten

Weitere Lösungen sind möglich.

- e) Durch die Verwendung standardisierter Testverfahren und -methoden wie z. B. automatisierter Regressionstests, kann sichergestellt werden, dass nach jeder neuen Softwareversion detailliert nach Fehlern gesucht wird.

Um eine bestmögliche Qualität während der Entwicklung zu gewährleisten, sollten des Weiteren auch anerkannte Softwareentwicklungsmodelle wie z. B. SCRUM oder KANBAN verwendet werden.

Weitere beispielhafte Maßnahmen zur Qualitätssicherung:

- Einhaltung der gesetzlichen Vorgaben zum Datenschutz (BDSG, DSGVO)
- Detaillierter Abgleich zwischen den Funktionalitäten der Software und dem Lasten- bzw. Pflichtenheft
- Detaillierter und permanenter Abgleich des definierten Soll-Zustandes und dem Ist-Zustand

Weitere Lösungen sind möglich.

2. Aufgabe

Qualitätsmerkmale gemäß ISO/IEC 9126:

Funktionalität

Eine Reihe von Attributen, die sich auf das Vorhandensein einer Reihe von Funktionen und deren spezifische Eigenschaften beziehen. Die Funktionen sind diejenigen, die erklärte oder implizierte Bedürfnisse befriedigen: Angemessenheit, Exaktheit, Interoperabilität, Sicherheit und Einhaltung der Funktionalität.

Zuverlässigkeit

Eine Reihe von Attributen, die sich auf die Fähigkeit einer Software beziehen, ihr Leistungsniveau unter bestimmten Bedingungen über einen bestimmten Zeitraum aufrechtzuerhalten: Ausgereiftheit, Fehlertoleranz, Wiederherstellbarkeit und Einhaltung der Zuverlässigkeitsanforderungen.

Benutzerfreundlichkeit

Eine Reihe von Attributen, die sich auf den für die Nutzung erforderlichen Aufwand und die individuelle Bewertung dieser Nutzung durch eine bestimmte oder implizite Gruppe von Benutzern auswirken: Verstehbarkeit, Erlernbarkeit, Bedienbarkeit, Anziehungskraft und Übereinstimmung mit der Gebrauchstauglichkeit.

Effizienz

Eine Reihe von Attributen, die sich auf das Verhältnis zwischen dem Leistungsniveau der Software und der Menge der verwendeten Ressourcen unter bestimmten Bedingungen beziehen: Zeitverhalten, Ressourcennutzung und Einhaltung der Effizienz.

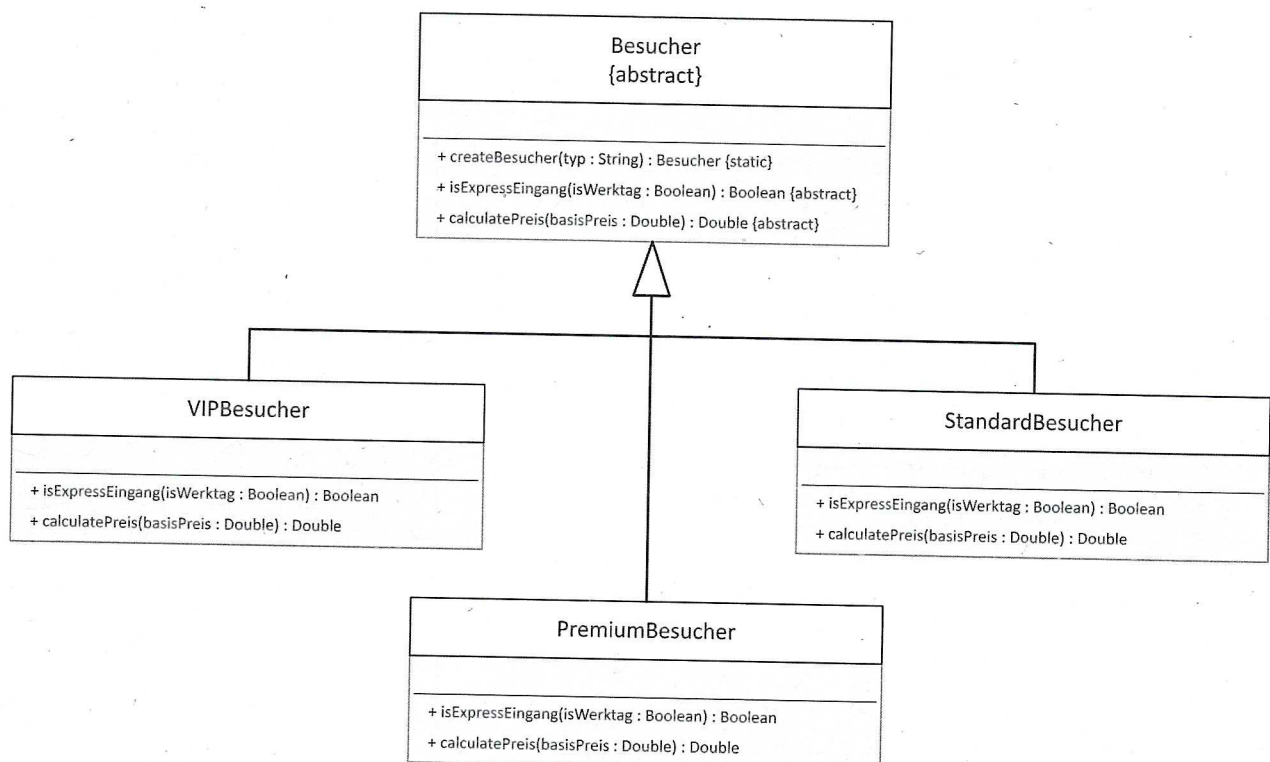
Portabilität

Eine Reihe von Attributen, die sich auf die Fähigkeit einer Software beziehen, von einer Umgebung in eine andere übertragen zu werden: Anpassungsfähigkeit, Installierbarkeit, Koexistenz, Ersetzbarkeit und Einhaltung der Portabilität.

ISO/IEC 9126 wurde inzwischen durch ISO/IEC 25000 ersetzt, Qualitätsmerkmale gemäß ISO/IEC 25000 sind demnach auch als richtig zu bewerten.

- ba) Polymorphie (Vielgestaltigkeit) ist ein zentrales Prinzip der objektorientierten Programmierung, das besonders im Kontext der Vererbung auftritt. Es beschreibt, dass Methoden mit identischer Signatur unterschiedliche Resultate liefern können. Erst zur Laufzeit wird entschieden, welche der Methoden für das jeweilige Objekt verwendet wird (dynamisches Binden). Man unterscheidet in statische (überladene Methoden) und dynamische Polymorphie (überschriebene Methoden).

bb)



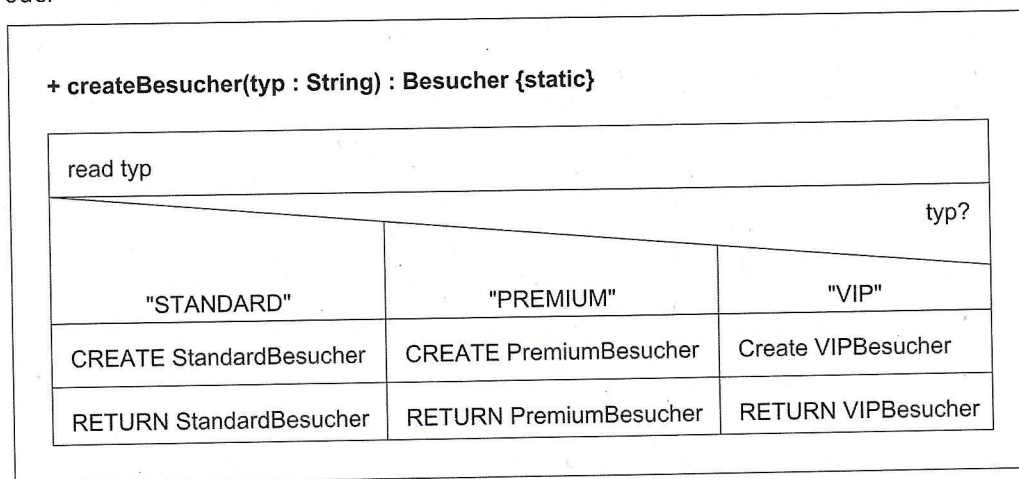

```

bc) public abstract class Besucher {
    //...
}

public class BesucherFactory {
    public static Besucher createBesucher(String typ) {
        switch (typ) {
            case "STANDARD":
                return new StandardBesucher();
            case "PREMIUM":
                return new PremiumBesucher();
            case "VIP":
                return new VIPBesucher();
        }
    }
}

```

oder



Ähnliche Lösungen sind möglich.

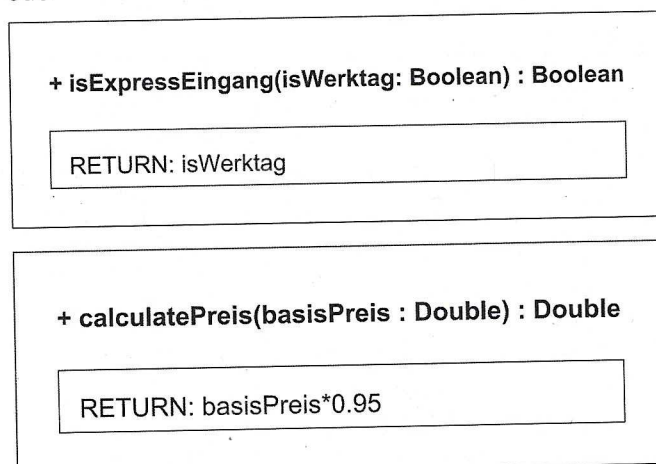
```

bd) @Override
    public Boolean isExpressEingang (Boolean isWerktag) {
        return isWerktag;
    }

    @Override
    public Double calculatePreis (Double basisPreis) {
        return basisPreis*0.95;
    }

```

oder



3. Aufgabe

aa) $C3 = A2 + 1$

ab) $E3 = \text{VLOOKUP}(C3; \text{Wetterdaten!B3:D17}; 3; \text{FALSE})$
oder
 $E3 = \text{SVERWEIS}(C3; \text{Wetterdaten!B3:D17}; 3; \text{FALSCH})$

ac) $F3 = H3 * \text{IF}(\text{WEEKDAY}(C3) \leq 5; 0,9; 1)$
oder
 $F3 = H3 * I3$

Mit der Formel $= \text{IF}(\text{WEEKDAY}(C3) \leq 5; 0,9; 1)$ in I3

Alternativ kann auch „WOCHENTAG“ und „WENN“ verwendet werden.

ba) Tabellenkalkulationsprogramme können z. B. auch für die nachfolgenden Einsatzmöglichkeiten genutzt werden:

Budgetplanung:

Verwaltung von Einnahmen und Ausgaben sowie Erstellung von Budgetplänen.

Datensammlung und -analyse:

Verwendung von Tabellenkalkulationsprogrammen zur Datenerfassung und -analyse in verschiedenen Bereichen, wie z. B. Wissenschaft, Forschung, Finanzen und Marketing.

Projektmanagement:

Verfolgung von Fortschritten, Verwaltung von Ressourcen und Überwachung von Meilensteinen bei Projekten.

Zeitplanung:

Erstellung von Zeitplänen für Projekte, um sicherzustellen, dass sie rechtzeitig abgeschlossen werden.

Personalplanung:

Verwaltung von Mitarbeiterdaten, z. B. Arbeitszeiten, Gehältern und Urlaubszeiten.

Inventar- und Bestandsmanagement:

Verwaltung von Inventar und Bestand für Unternehmen, z. B. Tracking von Produktions- oder Lagerbeständen.

Kundenbeziehungsmanagement:

Verwaltung von Kunden- und Verkaufsdaten für Unternehmen, z.B. Tracking von Verkaufszahlen und Kundendaten.

Weitere Lösungen sind möglich.

bb) Einige Probleme beim Einsatz von Tabellenkalkulationsprogrammen sind:

Fehleranfälligkeit:

Durch manuelle Eingabe und Bearbeitung der Daten können leicht Fehler auftreten, die schwer zu finden sein können.

Mangelnde Sicherheit:

Es besteht ein höheres Risiko für unerlaubten Zugriff, da Tabellenkalkulationsprogramme oft auf lokalen Computern gespeichert werden und nicht über die gleichen Sicherheitsmaßnahmen wie andere Anwendungen verfügen.

Unzureichende Datenvalidierung:

Ohne eine angemessene Überprüfung können Benutzer falsche Daten in die Tabellenkalkulation einfügen, was zu falschen Ergebnissen führen kann.

Schwierigkeiten bei der Zusammenarbeit:

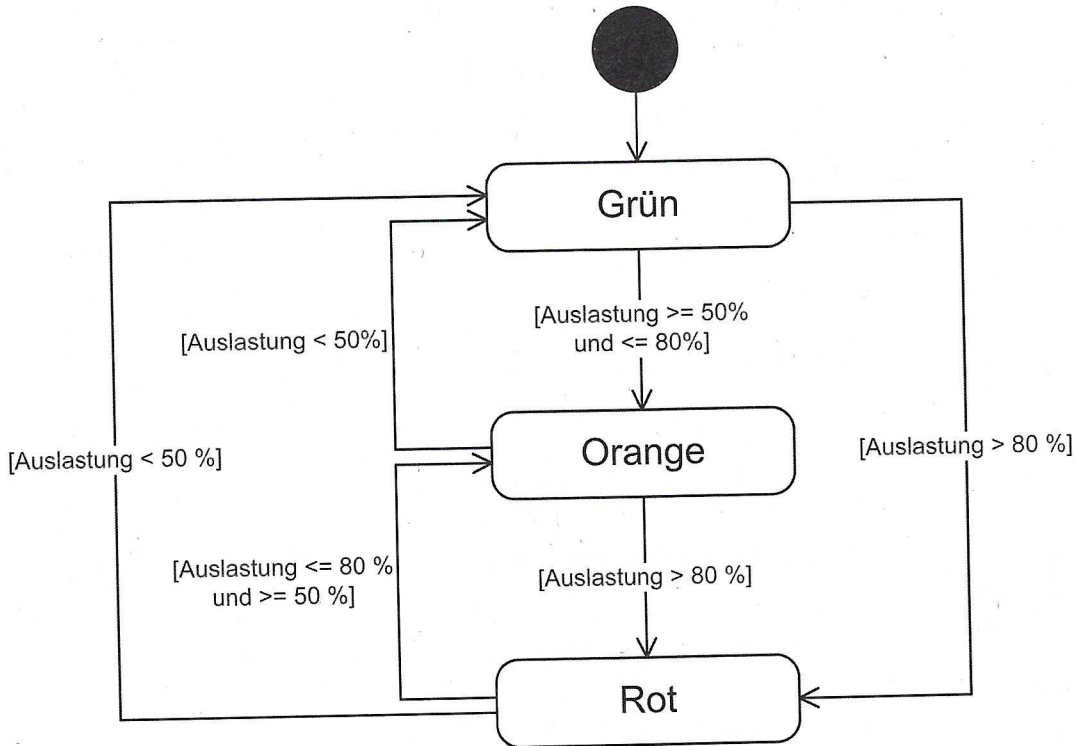
Bei der Zusammenarbeit an einer Tabellenkalkulation kann es zu Problemen kommen, wenn mehrere Personen gleichzeitig Änderungen vornehmen oder unterschiedliche Versionen der Tabelle bearbeiten.

Mangelnde Skalierbarkeit:

Tabellenkalkulationsprogramme sind möglicherweise nicht geeignet, um mit großen Datenmengen umzugehen oder komplexe Berechnungen durchzuführen.

Weitere Lösungen sind möglich.

c)



4. Aufgabe

- a) Für die sichere Übertragung von Daten zwischen den batteriebetriebenen Sensoren und dem Server in einem lokalen Funknetzwerk empfiehlt es sich, eine symmetrische Verschlüsselung wie AES oder ChaCha20 zu verwenden. Da diese Verfahren besonders ressourcenschonend, leicht und schnell sind.

Zur Authentifizierung der Sensoren und des Servers kann ein Pre-Shared Key (PSK) verwendet werden, der eine einfache Möglichkeit zur Gewährleistung von Vertraulichkeit und Integrität der Daten während der Übertragung bietet. Hierdurch muss keine komplexe Public-Key-Infrastruktur (asymmetrische Verschlüsselung) eingesetzt werden.

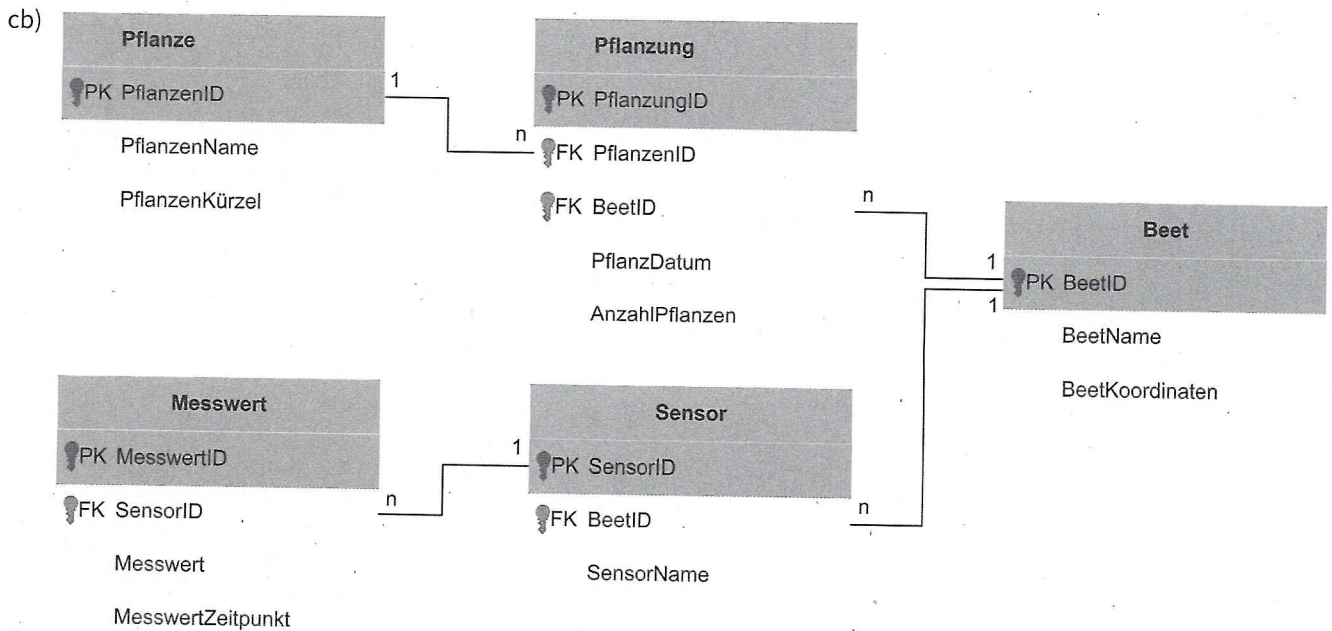
- b) Da die Daten, die zwischen den Sensoren und dem Server übertragen werden, keine besonderen Anforderungen an die Vertraulichkeit erfüllen müssen, kann das Signieren der Daten eine geeignete Alternative sein. Insbesondere da im Gegensatz zur Verschlüsselung weniger Rechenaufwand betrieben werden muss und hierdurch Energie eingespart werden kann.

Durch das Signieren würde sichergestellt werden, dass die Daten von einer vertrauenswürdigen Quelle stammen und unterwegs nicht verändert wurden. Es wird also vor allem die Authentizität und Integrität der Daten gewährleistet.

- ca) Wenn zum Beispiel ein Beet umbenannt wird und in einer anderen Tabelle auf den ursprüngliche Beet-Namen verwiesen wurde, kann es zu inkonsistenten Daten kommen, wenn der neue Beet-Name nicht in allen betroffenen Tabellen aktualisiert wird.

Dies kann dazu führen, dass Berichte oder Abfragen fehlerhafte Ergebnisse liefern oder dass bestimmte Aktionen, die auf das Beet bezogen sind, nicht korrekt ausgeführt werden.

Es handelt sich hierbei um eine Redundanzanomalie (speziell Änderungsanomalie).



Weiterführende Informationen:

Nachfolgend einige kurze Regeln für die Erstellung eines ER-Diagramms (Entity-Relationship-Diagramm):

1. Identifiziere die relevanten Entitäten, d.h. die Hauptobjekte, die im System modelliert werden sollen.
2. Definiere die Beziehungen zwischen den Entitäten, um die Verbindungen und Abhängigkeiten zwischen ihnen darzustellen.
3. Verwende Rechtecke, um Entitäten darzustellen, und Linien, um die Beziehungen zwischen ihnen zu verbinden.
4. Kennzeichne die Schlüsselattribute für jede Entität, um eindeutige Identifikation zu ermöglichen.
5. Definiere Kardinalitäten, um die Anzahl der Beziehungen zwischen Entitäten anzugeben (z. B. 1:1, 1:n, n:m).
6. Dokumentiere die Attribute für jede Entität, um ihre Eigenschaften und Bedeutungen festzuhalten.
7. Halte das Diagramm übersichtlich und gut strukturiert, um eine klare Darstellung der Datenmodellierung zu gewährleisten.
8. Beachte die Konventionen und Standards, die für ER-Diagramme gelten, um eine konsistente und verständliche Darstellung zu gewährleisten.

Diese Regeln dienen als allgemeine Richtlinien für die Erstellung eines ER-Diagramms. Es ist wichtig, den spezifischen Anwendungsfall und die Anforderungen des Datenmodells zu berücksichtigen, um anschließend das Diagramm entsprechend anzupassen.

