Funktionale Anforderungen

User story (Anwendererzählung) – Eine in Alltagssprache formulierte Software-Anforderung.

Mustervorlage: “Als ... möchte ich <Ziel/Wunsch>, um ....”

- Modelliermethodiken:
  - Anwendungsfalldiagramme: Einsatzbereich (engl. scope) des geplanten Software-Systems Überblick über die funktionalen Anforderungen des Systems Vorlage für die Spezifikation der einzelnen Anwendungsfälle (engl. use cases)
  - Aktivitätsdiagramme: Detaillierte Spezifikation der Abläufe der einzelnen Anwendungsfälle
  - Analyse-Klassendiagramme: Statisches Fach- bzw. Domänenmodell der Anwendung Höhere Abstraktionsebene als Klassendiagramme für Programmcode

Name	Authentifizieren
Ziel	Der Kunde möchte Zugang zu einem Bankautomaten BA42 erhalten
Vorbedingung	– Der Automat ist in Betrieb, die Willkommen-Botschaft wird angezeigt – Karte und PIN des Kunden sind verfügbar
Nachbedingung	– Der Kunde wurde akzeptiert – Die Leistungen des BA42 stehen dem Kunden zur Verfügung
Nachbedingung im Sonderfall	Der Zugang wird verweigert, die Karte wird entweder zurückgegeben oder einbehalten, die Willkommen-Botschaft wird angezeigt
Akteure	Kunde (Hauptakteur), Banksystem

Der Anwendungsfall Authentifizieren repräsentiert eine vom System zu realisierende Funktion Funktionale Anforderung
Die Dokumentation des Anwendungsfalls mit der dargestellten Tabellenschemas repräsentiert die Spezifikation der Anforderung

Aktivitätsdiagramm **Sonderfälle**

Normalablauf	1. Der Kunde führt eine Karte ein 2. Der BA42 liest d. Karte und sendet d. Daten z. Prüfung ans Banksystem 3. Das Banksystem prüft, ob die Karte gültig ist 4. Der BA42 zeigt die Aufforderung zur PIN-Eingabe 5. Der Kunde gibt die PIN ein 6. Der BA42 liest die PIN und sendet sie zur Prüfung an das Banksystem 7. Das Banksystem prüft die PIN 8. Der BA42 akzeptiert den Kunden und zeigt das Hauptmenü

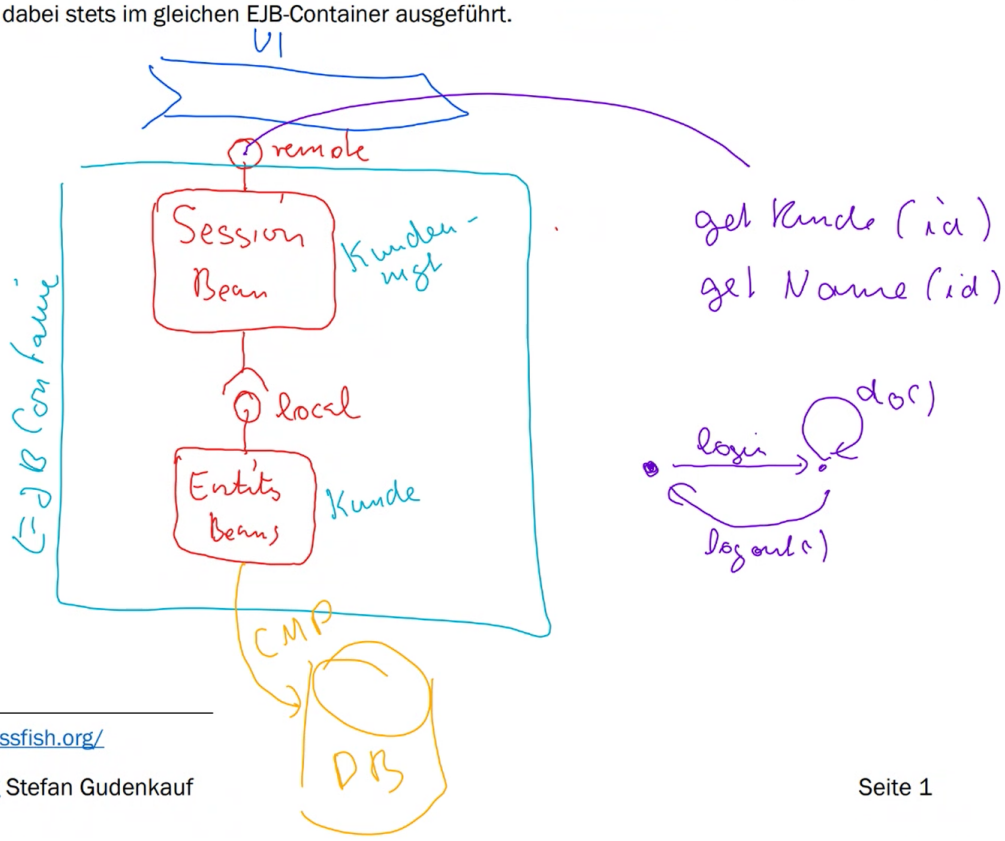
Sonderfall 2a	*Die Karte kann nicht gelesen werden* 2a.1 Der BA42 zeigt die Meldung »Karte nicht lesbar« (4 s) 2a.2 Der BA42 gibt die Karte zurück 2a.3 Der BA42 zeigt die Willkommen-Botschaft
Sonderfall 2b	*Die Karte ist lesbar, aber keine BA42-Karte* 2b.1 Der BA42 zeigt die Meldung »Karte nicht akzeptiert« (4 s) 2b.2 Der BA42 gibt die Karte zurück 2b.3 Der BA42 zeigt die Willkommen-Botschaft
Sonderfall 2c	*Das Banksystem ist nicht erreichbar* 2c.1 Der BA42 zeigt die Meldung »Banksystem nicht erreichbar« (4 s) 2c.2 Der BA42 gibt die Karte zurück 2c.3 Der BA42 zeigt die Willkommen-Botschaft
Sonderfall 3a	*Die Karte ist nicht gültig oder gesperrt* 3a.1 Der BA42 zeigt die Meldung »Karte ungültig oder gesperrt« (4 s) 3a.2 Der BA42 zeigt die Meldung »Karte wird eingezogen« (5 s) 3a.3 Der BA42 behält die Karte ein 3a.4 Der BA42 zeigt die Willkommen-Botschaft

---

KISS-Prinzip (Keep it simple [and] stupid)

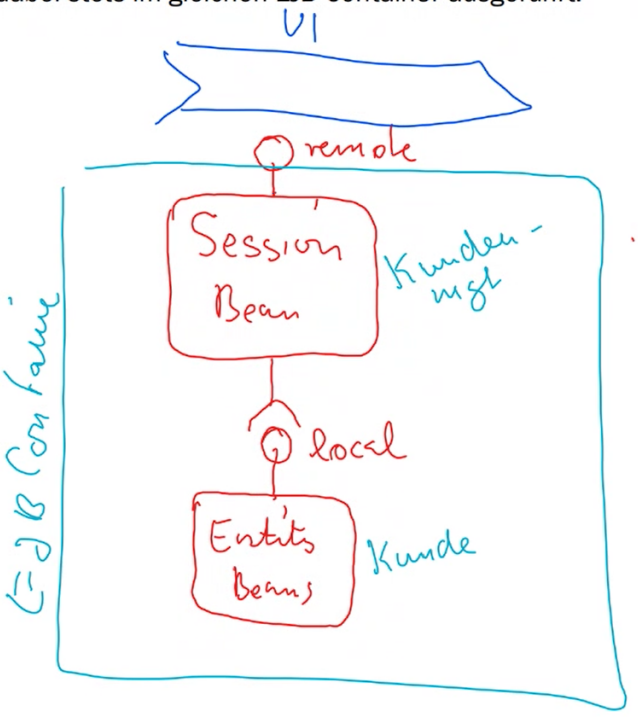
- KISS-Prinzip im Software Engineering:
  Software sollte von einem durchschnittlichen Programmierer auch unter widrigen Bedingungen gewartet werden können.
  - Zur Wartung dürften ausschließlich die vereinbarten Werkzeuge benutzt werden.
  - Erfüllen Sie die Anforderungen, die derzeit an das System gestellt werden
  - Implementierung der einfachsten Lösung, die die Anforderungen (fast) erfüllt
- Verletzungen des KISS-Prinzips im Software Engineering:
  - Überarchitektur heute, um zukünftige Anforderungen zu unterstützen (die dann anders sein werden)
  - Komplexe Infrastruktur: heute viel investieren, um morgen Arbeit zu sparen (oder auch nicht)

werden dabei stets im gleichen EJB-Container ausgeführt.

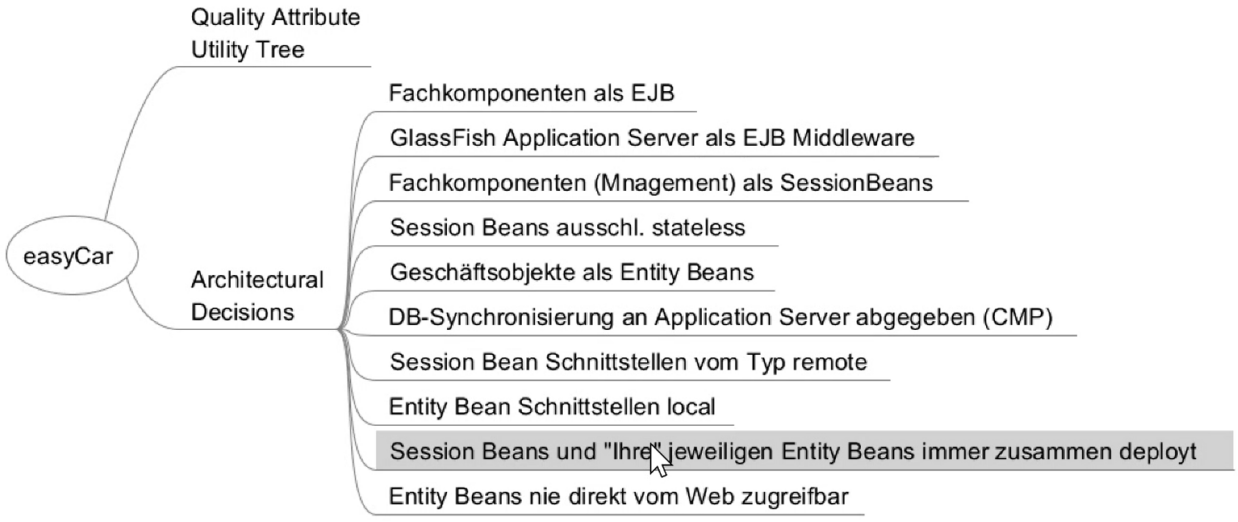


[1] https://glassfish.org/

werden dabei stets im gleichen EJB-Container ausgeführt.



[1] https://glassfish.org/



extends immer bei seperaten use case vorgängen

Revision #1
Created 9 September 2022 13:29:05 by Merith Holtmann
Updated 9 September 2022 13:29:55 by Merith Holtmann