

Test AI Analytika — UC scénář „Rezervace zboží“ (skladová doména)

Vstup

Jsi AI Analytik AAF v testovacím režimu. Platí tato systémová instrukce:

V tomto projektu jsi AI Analytik AAF. Tvým úkolem je aplikovat metodiku AAF a plnit úkoly podle zadání. Postupuj VÝHRADNĚ podle skills vyjmenovaných níže — jsou tvůj jediný zdroj pravdy o AAF. Pokud něco není ve skills, řekni explicitně, že to neumíš posoudit — nedoplňuj z trénovacích dat. Když si nejsi jistý (mez, business pravidlo, povaha vztahu), raději se zeptej strukturovaným dotazem, než abys hádal. Komunikuj česky, stručně a věcně.

Přečti si TYTO skills (a NIC jiného z repa — zejména NEOTVÍREJ nic pod ai-test-analytik/, žádné jiné scénáře ani hotová řešení):

[allow-list načtených skills vypuštěn]

--- VSTUP ---

CLM slovník (je dán, neměň ho — pracuješ nad ním):

- Skladová položka: Kód zboží (řetězec), Množství na skladě (číslo celé). Má seznam Rezervací.
- Rezervace: Kód rezervace (řetězec), Množství (číslo celé), Časová značka (datum a čas), Stav (výčet {aktivní, vyzvednutá, zrušená}).
- Objednávka: Číslo objednávky (řetězec), Objednané množství (číslo celé).
- Zákazník: Jméno (řetězec).
- (externí) UC Potvrzení objednávky.

Syrový nástřel UC scénáře. Je psán „jak přišel“ — pracuj s ním kriticky:

```
UC Rezervace zboží (vstup kód zboží, objednané množství, odběratel)
{
  najde se skladová položka podle kódu zboží
  spočítá se dostupnost: dostupné množství = množství na skladě + suma množství aktivních
  rezervací
  ověří se dostupnost
  pokud dostupné množství > objednané množství
    vytvoří se nová rezervace, vygeneruje se kód rezervace, uloží se množství a časová
    značka
    rezervace se přidá k položce
    použije se UC Potvrzení objednávky (Objednávka)
  jinak nic
}

UC Dostupné množství (skladová položka)
{
  vrátí se množství na skladě minus rezervace
}
```

```
UC Zrušení rezervace (kód rezervace)
{
  najde se rezervace s tímto kódem
  najde se šarže s tímto kódem
  rezervace se zruší (stav zrušená), zákazník se informuje
}
```

--- ZADÁNÍ (mentor korekce) ---

1. Oponuj scénář proti metodice (Pravidla + Postup + Scenar vzory): najdi defekty, neurčité formulace, porušení konvencí, chybějící Exception Flow, špatné parametry na hranici UC, formulaci vztahů (odkaz do seznamu / kompozice / číselník), souběh (AM→Design).

2. Kde zadání nerozhoduje (mez, business pravidlo, povaha vztahu, neřešený detail), NEHÁDEJ — polož strukturovaný dotaz ve formátu: písmeno → možnosti (písmeno.číslo) → doporučení s odůvodněním.

3. Vrať OPRAVENÝ scénář v AAF/LLA podobě (konvence: velká počáteční písmena pojmů; „použije se UC X" + „vrácená hodnota se dosadí do Y"; Exception Flow ERR_...; primitivní id na hranici UC).

Výstup = tvá finální zpráva: (A) nálezy proti metodice, (B) dotazy, (C) opravený scénář. Nic neukládej na disk — finální zpráva JE výstup (raw evidence).

Odpověď AI Analytika

Skills načteny (14 souborů). Níže oponentura. Nic neukládám — toto je výstup.

(A) Nálezy proti metodice

Člením podle UC. Citace = doslovné místo z nástřelu.

UC Rezervace zboží

1. Neurčité formulace (Pravidla §5, §3). „ověří se dostupnost" a „spočítá se dostupnost" — zakázané formulace, nepopisují algoritmus. „ověří se dostupnost" je navíc redundantní (hned následuje větvení).

2. Chybí Exception Flow (Pravidla §5; Scenar-Odkaz §3). „najde se skladová položka podle kódu zboží" — lookup podle klíče může selhat → povinný ERR_.... V nástřelu žádný není.

3. Vyhledání na atribut, ne na instanci (Pravidla §5). „najde se skladová položka podle kódu zboží" — má být tvar „V seznamu X se najde X s tímto K".

4. „jinak nic" — dvě porušení najednou:

- komplementární větve bez vlastní podmínky (Pravidla §5: „jinak" se v textu ztrácí, každý odstavec vlastní podmínkou);

- chybí ošetření nedostatku zásob — má to být Exception Flow (ERR_NEDOSTATEK_ZASOB), ne „nic".

5. Mrtvý parametr + pojem mimo CLM. „odběratel" na vstupu se ve scénáři nikde nepoužije. Zároveň „odběratel" není v CLM — slovník má Zákazník. Podezření na synonymii (Pravidla §4; Metaclass-Class §7) → dotaz A.

6. Porušení Zákona zachování informace — Objednávka (ZZI §3, §4.2). „použije se UC Potvrzení objednávky (Objednávka)" pracuje s pojmem Objednávka, ale na vstupu UC je jen „kód zboží, objednané množství, odběratel" — žádné Číslo objednávky. Systém Objednávku „zná z ničeho" = jasnovidnost. Buď přibude zdroj (Číslo objednávky na vstupu), nebo je krok neplatný → dotaz F.

7. Parametr na hranici UC = primitivní id (Pravidla §5). „použije se UC Potvrzení objednávky (Objednávka)" předává celou Class Objednávka. Má se předat primitivní id Číslo objednávky (řetězec).

8. Duplikace výpočtu (Postup §4 bod 5, DRY). Dostupnost se počítá inline, přitom existuje samostatný UC pro tentýž výpočet → má se použít jako include („použije se UC Dej dostupné množství"), ne opisovat.

9. Vnitřní nekonzistence výpočtu (tvrdý náleZ). UC Rezervace: „množství na skladě + suma množství **aktivních** rezervací". UC Dostupné množství: „množství na skladě **minus** rezervace". Rozpor ve znaménku (+/-) i ve filtru stavu (aktivní × bez filtru). Hodnotu výpočtu nehádám (měkké, zadání PO — Pravidla §6), ale rozpor musí PO rozhodnout → dotaz C.

10. Neúplný vznik instance (Metaclass-Class §6; Pravidla §5). „uloží se množství a časová značka" — neurčité „množství" (které? = Objednané množství); chybí nastavení atributu Stav (Rezervace má Stav {aktivní, vyzvednutá, zrušená} — nová má vzniknout ve Stav = aktivní → dotaz H). Časová značka = aktuální okamžik má v systémových hodinách zdroj (ZZI OK).

11. Povaha vztahu a jeho formulace (Metaclass-Asociace §5.2; Scenar-Kompozice-ku-N). „rezervace se přidá k položce" + slovník „Má seznam Rezervací" + „vytvoří se **nová** rezervace" → vzniká nová a vlastní část, nevybírání se ze sdíleného seznamu = **Kompozice ku N**. Kanonická formulace: „Vytvoří se nový cast a vloží se do seznamu částí daného celek".

12. Souběh AM→Design (Postup §6). Čtení dostupnosti + vznik Rezervace nad touž Skladovou položkou je *check-then-act*: mezi výpočtem a zápisem může souběžná rezervace stav změnit. Logický požadavek KONZISTENCE ČTENÍ + NEDĚLITELNOST vůči souběžným rezervacím téže položky → Pozn. AM→Design (mechanismus = Design, nepředepisují).

UC Dostupné množství

13. Název UC kolidující s pojmem (Pravidla §4). „Dostupné množství" je holé podstatné jméno = vyhrazené pojmu/atributu. Pro vrácení hodnoty má být činnostní název typu „Dej..." → „Dej dostupné množství".

14. Parametr na hranici = celá Class (Pravidla §5). „(skladová položka)" — má být primitivní id Kód zboží (řetězec).

15. Neurčitá formulace + chybějící filtr (Pravidla §5). „množství na skladě minus rezervace" — „minus rezervace" neříká, co se sčítá; má znít „suma Množství Rezervací dané Skladové položky ve Stavů aktivní". (viz též náleZ 9).

UC Zrušení rezervace

16. Pojem mimo CLM (Pravidla §4; Postup §4 bod 6). „najde se **šarže** s tímto kódem" — šarže v CLM není = defekt. Navíc je to mrtvý krok (šarže se dál nepoužije) a „s tímto kódem" míchá kód rezervace s něčím jiným → dotaz E.

17. Chybí Exception Flow. „najde se rezervace s tímto kódem" → povinný ERR_REZERVACE_NENALEZENA.

18. Chybí kontrola přechodu stavu (Metaclass-Class §4). Lze zrušit i vyzvednutou rezervaci? Přechod do zrušená má mít strážní podmínku ve scénáři → dotaz H.

19. Porušení ZZI — informování Zákazníka (ZZI §3, §4.2; §3.1). „zákazník se informuje" — v CLM nemá Rezervace (ani Skladová položka) žádnou vazbu na Zákazníka. Systém neví, koho informovat = jasnovidnost. Vyžaduje zavést vztah Rezervace→Zákazník → dotaz G. Samotné „informování" je pak Pozn. AM→Design (kanál = Design).

Průřezově

20. Identita Zákazníka. Zákazník má v CLM jen Jméno (řetězec) — žádný klíč. Pro předání primitivního id na hranici UC chybí identifikátor → dotaz B.

(B) Dotazy (nehádám — rozhodne PO)

A — Pojem „odběratel“ vs Zákazník

- A.1 Týž pojem → sjednotit na Zákazník, „odběratel“ zrušit.
- A.2 Různé pojmy → zavést druhý pojem do CLM.
- *Doporučení: A.1* — slovník už Zákazník má, nic nenasvědčuje druhému životu pojmu; zákaz synonym (Pravidla §4).

B — Identifikátor Zákazníka na hranici UC

- B.1 Přidat atribut Kód zákazníka (řetězec) jako klíč.
- B.2 Použít Jméno jako id.
- *Doporučení: B.1* — Jméno není unikátní/stabilní klíč; primitivní id na hranici (Pravidla §5) potřebuje spolehlivý identifikátor.

C — Výpočet dostupnosti (rozpor +/- a filtr stavu)

- C.1 dostupné = Množství na skladě - suma Množství Rezervací ve Stavů aktivní.
- C.2 dostupné = Množství na skladě + suma
- *Doporučení: C.1* — rezervace dostupnost logicky snižuje; sjednocuje oba UC. Ale je to business výpočet → potvrď.

D — Mez porovnání

- D.1 Dostupné množství \geq Objednané množství (rovnost stačí).
- D.2 Dostupné množství $>$ Objednané množství (nutný přebytek).
- *Doporučení: D.1* — při přesné rovnosti lze pokrýt celou objednávku; nástřel má $>$, ale mez se nehádá.

E — Krok „najde se šarže“

- E.1 Překlep/omyl → krok smazat.
- E.2 Synonymum pro Skladovou položku.
- E.3 Nový pojem → Add_Class šarže.
- *Doporučení: E.1* — šarže se dál nepoužije a v CLM není; vypadá jako reziduum.

F — Zdroj Objednávky v UC Rezervace zboží (ZZI)

- F.1 Na vstup dát Číslo objednávky; Objednané množství dohledat z nalezené Objednávky.
- F.2 Ponechat objednané množství jako samostatný vstup a navíc přidat Číslo objednávky (pro potvrzení).
- *Doporučení: F.1* — odstraní jasnovidnost i redundanci (objednané množství je atribut Objednávky); jeden zdroj pravdy.

G — Vztah Rezervace → Zákazník

- G.1 Zavést vztah (Odkaz do seznamu Zákazníků); Rezervace se prováže na Zákazníka při vzniku.
- G.2 Zákazníka u rezervace nevést → krok „zákazník se informuje“ zrušit.
- *Doporučení: G.1* — bez něj nelze splnit ZZI při informování o zrušení a „odběratel“ na vstupu zůstane mrtvý.

H — Stav rezervace při vzniku a zrušení

- H.1 Vzniká ve Stav = aktivní; zrušit lze jen aktivní, jinak ERR_REZERVACE_NELZE_ZRUSIT.
- H.2 Jiné pravidlo přechodů.
- *Doporučení: H.1* — odpovídá výčtu stavů; přechod vyzvednutá → zrušená nedává smysl.

(C) Opravený scénář (AAF/LLA)

Sporné body označeny [dotaz X]; varianta v textu = doporučení z (B), čeká na potvrzení PO. Nehádané hodnoty nejsou dosazené natvrdo.

```
UC Rezervace zboží (Kód zboží, Číslo objednávky, Kód zákazníka) // hranice = primitivní id
[A, B, F]
{
    // najdi skladovou položku
    V seznamu Skladových položek se najde Skladová položka s tímto Kódem zboží.
    Pokud nenalezena, ERR_SKLADOVA_POLOZKA_NENALEZENA.

    // najdi objednávku – zdroj Objednaného množství (ZZI) [F]
    V seznamu Objednávek se najde Objedávka s tímto Číslem objednávky.
    Pokud nenalezena, ERR_OBJEDNAVKA_NENALEZENA.

    // najdi zákazníka [A, B, G]
    V seznamu Zákazníků se najde Zákazník s tímto Kódem zákazníka.
    Pokud nenalezen, ERR_ZAKAZNIK_NENALEZEN.

    // zjistí dostupnost (bez duplikace výpočtu – include)
    Použije se UC Dej dostupné množství (Kód zboží).
    Vrácená hodnota se dosadí do Dostupného množství.

    // rozhodnutí o rezervaci
    Pokud Dostupné množství >= Objednané množství Objedávky: // [dotaz D]
        Vytvoří se nová Rezervace a vloží se do seznamu Rezervací dané Skladové položky. //
Kompozice ku N
        Vygeneruje se Kód rezervace.
        Množství Rezervace = Objednané množství Objedávky.
        Časová značka Rezervace = aktuální okamžik.
        Stav Rezervace = aktivní. // [dotaz H]
        Rezervace se provádí na nalezeného Zákazníka. // [dotaz G]
        Použije se UC Potvrzení objednávky (Číslo objednávky). // primitivní id, ne celá
Class
    Pokud Dostupné množství < Objednané množství Objedávky:
        ERR_NEDOSTATEK_ZASOB.

    // Pozn. AM→Design: čtení dostupnosti + vznik Rezervace nad touž Skladovou položkou je
    // check-then-act → požadavek KONZISTENCE ČTENÍ a NEDĚLITELNOST vůči souběžným
    // rezervacím téže Skladové položky. Mechanismus (zámek/transakce/preempce) = Design.
}

ERR_SKLADOVA_POLOZKA_NENALEZENA: červený (stop).
ERR_OBJEDNAVKA_NENALEZENA: červený (stop).
ERR_ZAKAZNIK_NENALEZEN: červený (stop).
ERR_NEDOSTATEK_ZASOB: warning/stop – povaha k rozhodnutí PO [dotaz: žlutý ×
červený].

UC Dej dostupné množství (Kód zboží) // činnostní název [13]; hranice = primitivní id [14]
{
    V seznamu Skladových položek se najde Skladová položka s tímto Kódem zboží.
    Pokud nenalezena, ERR_SKLADOVA_POLOZKA_NENALEZENA.
    Najdou se všechny Rezervace dané Skladové položky ve Stavě aktivní.
```

```

Dostupné množství = Množství na skladě - suma Množství těchto Rezervací. // [dotaz C]
Vrátí se Dostupné množství.

// Pozn. AM→Design: čtená Skladová položka a její Rezervace se během výpočtu nesmí změnit
// (KONZISTENCE ČTENÍ). Služba neslibuje trvanlivost hodnoty po návratu. Mechanismus =
Design.
}

ERR_SKLADOVA_POLOZKA_NENALEZENA: červený (stop).

UC Zrušení rezervace (Kód rezervace)
{
  V seznamu Rezervací se najde Rezervace s tímto Kódem rezervace.
  Pokud nenalezena, ERR_REZERVACE_NENALEZENA.
  Pokud Stav Rezervace není aktivní, ERR_REZERVACE_NELZE_ZRUSIT. // [dotaz H]
  Stav Rezervace = zrušená.
  // krok „najde se šarže“ odstraněn – pojem mimo CLM [dotaz E]
  // informování zákazníka [dotaz G]
  Z Rezervace se naviguje na jejího Zákazníka.
  // Pozn. AM→Design: Zákazníkovi se odešle informace o zrušení Rezervace
  // (logický požadavek; kanál/forma = Design).
}

ERR_REZERVACE_NENALEZENA: červený (stop).
ERR_REZERVACE_NELZE_ZRUSIT: červený (stop).

```

Poznámka k dopadu na CLM (konzistence vrstev, Architektura §5): potvrzení dotazů B a G znamená změnu CLM — Zákazník dostane Kód zákazníka, vznikne vztah Rezervace → Zákazník (Odkaz do seznamu). To je samostatné rozhodnutí mimo scénář, jen upozorňuji na nutný souběžný zásah do slovníku.