

Test AI Analytika — UC scénář „Vygenerování čekající platby“

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):

- Platba – Class. Atributy: Kód rezervace (řetězec, unikátní), Částka (peněžní částka). Vazby: Odkaz do seznamu → Kino; seznam Stav platby (vzor Vlak má svoje vagóny).
- Stav platby – Class (vagón Vlaku, reifikovaný záznam změny stavu). Atribut: Timestamp (datum a čas). Vazba: Odkaz do seznamu → číselník Stav pro platby. (Platba ◆– N Stav platby.)
- Stav pro platby – číselník (Class). Atribut: Kód; hodnoty {čekající, zaplacená, stornovaná}.
- Kino – Class. Identifikace přes Identifikace kina.
- (externí) UC Verifikace kina – vstup Identifikace kina; vrací Kino; detail verifikace není řešen v této iteraci.

Syrový nástřel UC scénáře (pochází z původní dokumentace MTP, pre-AAF, styl SCRUM/UML). Je psán „jak přišel“ — pracuj s ním kriticky:

```
UC Vygenerování čekající platby (vstup Identifikace kina, částka) (výstup kód rezervace)
{
  provede se UC Verifikace kina {Identifikace kina} vrací se nalezené verifikované Kino (není
řešeno v tomto kroku iterace)
  Vytvoří se nový prvek Platba a naváže se na nalezené Kino. Vytvoří se nový kód rezervace a
vloží se spolu s prvkem přijatá částka do nového prvku Platba.
  Vytvoří se nový prvek Stav platby a přidá se do seznamu stavů platby daného prvku Platba,
vloží se do něj aktuální timestamp a prováže se na prvek číselníku Stav pro platby {kod = 1,
čekající}.
  Vrací se kód rezervace resp. chybový stav.
}
```

--- ZADÁNÍ (mentor korekce) ---

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

(A) Nálezy proti metodice

A1. Vstupní signatura UC nese nepřimitivní parametr.

Hlavička (vstup Identifikace kina, částka). Pravidlo Pravidla §5 (parametr na hranici UC = primitivní typ / id): na vstup patří primitivní identifikátor. Identifikace kina je v CLM zaveden jako identifikace přes níž externí UC vrací Kino — je to klíč, takže pravděpodobně OK jako id (řetězec/kód), ale není to ve vstupu deklarováno jako primitivní typ. Částka je peněžní částka (primitivní) — OK. Defekt je formulační: parametry nemají uvedenou povahu a Identifikace kina je třeba potvrdit jako primitivní klíč, ne jako Class.

A2. Volání externího UC není zapsáno rezervovanou formulací.

provede se UC Verifikace kina {...} vrací se nalezené verifikované Kino — pravidlo Pravidla §5 / Cinknutí §3 vyžaduje „**použije se UC X**" + „**vrácená hodnota se dosadí do Y**". Navíc se vrací **celá Class Kino** jako nález (to je v pořádku — vyhledání instance vrací instanci, Pravidla §5 „vyhledávání celé instance"), ale dosazení musí být explicitní.

A3. Chybí Exception Flow — žádný ERR_....

Scénář na konci zmiňuje „resp. chybový stav", ale nedefinuje žádnou kontrolu ani ERR_<název> (červený/žlutý). Pravidla §5. Minimálně: selhání verifikace kina (UC Verifikace kina nemusí Kino najít). Že „detail verifikace není řešen v této iteraci" neznamena, že se ignoruje jeho neúspěšný návrat — scénář musí ošetřit, co se stane, když se Kino nevrátí.

A4. Neurčitá / vágní formulace „resp. chybový stav".

Pravidla §3, §5 zakazují neurčitě formulace. „Vrací se kód rezervace resp. chybový stav" nepopisuje algoritmus — větvení musí mít vlastní podmínku a chybové stavy patří do Exception Flow, ne do alternativy návratové hodnoty.

A5. Vztah Platba ♦ — Stav platby: scénář používá Kompozici ku N správně intent, ale formulace není kanonická.

Vytvoří se nový prvek Stav platby a přidá se do seznamu stavů platby odpovídá intentu „přidání části do seznamu" (Scenar-Kompozice-ku-N §3, systém/import: „Vytvoří se nový cast a vloží se do seznamu částí daného celek."). Formulace je blízká, ale role nejsou dosazeny kanonicky a chybí návaznost na to, že **Stav platby je vagón složeného vzoru** (viz A6).

A6. Vazba Stav platby → Stav pro platby je Odkaz do seznamu (číselník) — formulace porušuje anti-pattern „prováže na id".

prováže se na prvek číselníku Stav pro platby {kod = 1, čekající}. Dvě věci:

- (a) Scenar-Odkaz-do-seznamu §2 anti-pattern: nevázat na F.K./id, ale **na prvek**. Zápis {kod = 1, ...} zavádí číselný kód 1, který v CLM **není** — číselník má atribut Kód a hodnoty {čekající, zaplacená, stornovaná}, žádné 1. To je porušení slovníku (Pravidla §4) — kod = 1 je „pojmem odnikud“ (číslo se v CLM nevyskytuje).
- (b) Kanonická formulace systém/import: „Podle tohoto kódu se v seznamu Stav pro platby najde Stav pro platby. Pokud nenalezen, ERR_...“ — zde je hodnota dána konstantou (čekající), takže lookup je podle konstantní hodnoty, ne podle přijatého kódu.

A7. Celý vztah Platba ♦ — Stav platby → Stav pro platby je složený vzor Vlak má svoje vagóny, ale scénář ho vede jako dvě nezávislé operace.

CLM to deklaruje explicitně (Platba ♦ — N Stav platby (Kompozice ku N) + Stav platby → Stav pro platby (Odkaz do seznamu), cast = source). To je přesně Pattern-Vlak-ma-svoje-vagony. Scénář by měl vznik nového Stavů platby a jeho provádění na číselník vést jako **jednu reifikovanou vazbu** (vznikne vagón XY = Stav platby, dosadí se Timestamp, prováže se na položku Y = Stav pro platby). Věcně to scénář dělá, ale nepojmenovává to jako složený vzor a Timestamp se plní „aktuální timestamp“ — viz A8.

A8. „Aktuální timestamp“ — Zákon zachování informace.

vloží se do něj aktuální timestamp. Zakon-zachovani-informace §4.2: systém pracuje s informací (aktuální čas) — zdroj musí být v modelu. Aktuální čas je systémové „čtení hodin“; to je hraniční, ale ve scénáři nemá deklarovaný kanál (není to vstup ani evidence). Buď je to legitimní systémové čtení času (a pak se to formuluje jako takové), nebo chybí zdroj. Spíš Pozn. AM→Design než tvrdá chyba — viz dotaz C/B2.

A9. Název UC — drobné.

Vygenerování čekající platby je činnostní podstatné jméno užitku (OK dle Use-Case-1-druhu §7, dokonce je to přímo příklad v tom skillu). Bez výhrad.

A10. Pojmy s velkým počátečním písmenem — nekonzistentní.

přijatá částka, kód rezervace se ve scénáři píší malými písmeny, ale jsou to atributy CLM (Částka, Kód rezervace). Pravidla §4: modelovaný pojem (Class i atribut) = velké počáteční písmeno.

A11. „prvek Platba“ / „prvek Stav platby“ — balast.

Pravidla §3 (žádný balast). Slovo „prvek“ před názvem Class je vata; stačí „vytvoří se nová Platba“.

(B) Dotazy

B. Exception Flow při selhání verifikace kina

Scénář deklaruje, že detail UC Verifikace kina „není řešen v této iteraci“, ale jeho návrat se ve scénáři používá.

Otázka, jak ošetřit neúspěšný návrat:

- B.1 — Verifikace kina **vždy vrátí Kino** (selhání řeší uvnitř sama, garantuje nennul návrat); náš scénář žádný ERR_ k verifikaci nepíše.
- B.2 — Verifikace kina **může nevrátit Kino**; náš scénář ošetří ERR_KINO_NEVERIFIKOVANO (červený, stop) a Platba nevznikne.
- B.3 — odloženo na pozdější iteraci; ve scénáři ponechat Pozn. o neřešeném větvení.
- Doporučení: **B.2**. Zákon zachování informace i Pravidla §5 (povinný Exception Flow u lookupu, který může selhat — Scenar-Odkaz-do-seznamu §3) říkají, že lookup/verifikace, jejíž návrat řídí další běh, musí mít ošetřené NENALEZENO. „Neřešený detail uvnitř“ se týká vnitřku Verifikace, ne našeho ošetření jejího návratu.

C. Zdroj aktuálního času pro Timestamp

- C.1 — Aktuální čas je **systémové čtení hodin** (legitimní zdroj, nevyžaduje vstup); ve scénáři ponechat jako „dosadí se aktuální datum a čas systému“, případně Pozn. AM→Design.
- C.2 — Timestamp **přichází jako vstup** UC (musel by být v signatuře) — pak porušení, protože v hlavičce není.

- C.3 — Timestamp se neplní při vzniku, plní ho jiný mechanismus.
- Doporučení: **C.1.** Čtení systémového času je standardní zdroj informace; označit poznámkou Pozn. AM→Design (mechanismus získání času = Design), logický požadavek „okamžik vzniku stavu“ zůstává na AM. Nehádám ale sám — Zakon-zachovani - informace mě nutí potvrdit, že systémové hodiny jsou v této doméně přijatý kanál.

D. Povaha a primitivní typ Identifikace kina na vstupu

- D.1 — Identifikace kina je **primitivní klíč** (řetězec/kód), který se předá UC Verifikace kina jako id; na vstupu UC je tedy primitivní typ (OK dle Pravidla §5).
- D.2 — Identifikace kina je **Class** (strukturovaný identifikátor) — pak porušení pravidla „primitivní id na hranici“ a je třeba ho rozložit / předat jen klíč.
- Doporučení: **D.1.** CLM ho popisuje jako „vstup Identifikace kina“ do externího UC vracejícího Kino, což odpovídá roli primitivního vyhledávacího klíče. Potvrdit povahu (řetězec).

(C) Opravený scénář (AAF / LLA)

Pozn.: opravený scénář předkládám pod **doporučenými** odpověďmi na B/C/D (B.2, C.1, D.1). Pokud autor rozhodne jinak, scénář se upraví.

```

UC Vygenerování čekající platby (vstup Identifikace kina : řetězec, Částka : peněžní částka)
(výstup Kód rezervace : řetězec)
{
  // verifikace kina
  použije se UC Verifikace kina (Identifikace kina); vrácená hodnota se dosadí do Kino.
  pokud se Kino nevrátí (neverifikováno), ERR_KINO_NEVERIFIKOVANO.

  // založení Platby
  vytvoří se nová Platba a prováže se na nalezené Kino.
  vygeneruje se nový Kód rezervace a dosadí se do Platby; do Platby se dosadí přijatá Částka.

  // založení čekajícího stavu (Vlak má svoje vagóny: Stav platby je vagón, Stav pro platby
je sdílená položka)
  vytvoří se nový Stav platby a vloží se do seznamu stavů dané Platby.
  do Stav platby se dosadí aktuální datum a čas (Pozn. AM→Design: mechanismus získání času =
Design).
  v seznamu Stav pro platby se najde Stav pro platby s hodnotou čekající; Stav platby se
prováže na nalezený prvek.

  // návrat
  vrátí se Kód rezervace dané Platby.
}

ERR_KINO_NEVERIFIKOVANO (červený, stop): Kino se podle Identifikace kina neverifikovalo;
Platba nevznikne, UC končí chybou.

```

Poznámky k opravě (mapování na nálezy):

1. Vstup/výstup uveden s primitivní povahou (A1, D.1).
2. Volání externího UC rezervovanou formulací „použije se UC X“ + „vrácená hodnota se dosadí do Y“ (A2).
3. Doplněn Exception Flow ERR_KINO_NEVERIFIKOVANO, červený/stop (A3, A4, B.2).
4. Odstraněn neexistující kod = 1; prováždání na číselník přes nalezený prvek podle hodnoty čekající (A6), bez F.K./id.
5. Pojmy velkými počátečními písmeny — Kód rezervace, Částka, Platba, Stav platby, Stav pro platby, Kino (A10).

6. Odstraněn balast „prvek“ (A11).

7. Vznik Stav platby a prováždání na číselník vedeno jako reifikovaná vazba složeného vzoru *Vlak má svoje vagóny* (A5, A7).

8. Timestamp ponechán jako systémové čtení času s Pozn. AM→Design (A8, C.1).

Co jsem **nehádal**: zda Verifikace kina může selhat (B), zda jsou systémové hodiny přijatý kanál (C), přesnou povahu Identifikace kina (D). Scénář výše stojí na doporučených variantách — čeká na potvrzení.