

Nefunkciniai (kokybiniai) reikalavimai sistemai [privaloma]

Visi reikalavimai yra privalomi, išskyrus tuos, kurie pažymėti „[papildoma]“. Kiekvienas privalomas reikalavimas vertinamas 0,25 balo.

1. [Concurrency] Vienas naudotojas gali dirbti su sistema (tame tarpe redagavimo/administravimo režimu) keliuose naršyklės languose (tiek „window“, tiek „tab“ prasme) tuo pačiu metu, naudodamas tą pačią paskyrą (account). {*use-case duomenų nesaugoti sesijoje; JSF: ViewScoped, ConversationScoped*}
2. [Short transactions] Jokia duomenų bazės transakcija (taip pat ir lentelės įrašų rašymas – angl. locking) negali būti ilgesnė nei interakcija su naudotoju (pvz.: pradedame transakciją, laukiame kol naudotojas užpildys/paredaguos Web formą, pabaigame transakciją). Jei DB transakcija naudojama, ji turi prasidėti ir pasibaigti tarp HTTP užklauskos (request) ir atsakymo (response) į *tą pačią* užklauską (nesvarbu, ar tai AJAX užklausa, ar ne). {*arba naudoti automatines transakcijas, arba įgyvendinti užtikrinimą, kad užklauskos pabaigoje būtų pabaigta užklauskos pradžioje pradėta transakcija; daugiau informacijos*}
3. [Optimistic locking] Keliems naudotojams (arba vienam keliuose naršyklės languose) tyčia/netyčia redaguojant tą patį DB objektą ir, kylant redaguojamų duomenų versijų konfliktui:
 - apie konfliktą turi būti perspėjamas naudotojas [privaloma],
 - naudotojui leidžiama pasirinkti, ar duomenis atsinaujinti, sulyginti ir pakartoti redagavimą, ar aklai užrašyti „ant viršaus“ [papildoma].{*JPA optimistic locking*}
4. [Security] Sistema turi būti apsaugota nuo SQL injection atakų. {*JPA queries with parameters / JDBC prepared statements*}
5. [Memory management] Parinkti techniniai projektiniai sprendimai turi būti orientuoti į taupų atminties (RAM) naudojimą. {*CDI/EJB component life-cycle management; use-case įgyvendinimui nenaudoti SessionScoped komponentų*}
6. [Cross-cutting functionality; [Extensibility-Glass-box extensibility](#)] Visi dalykinio funkcionalumo (angl. business logic) veiksmai, susiję su pinigais/apmokėjimais, privalo būti audituojami (fiksuojami faile arba DB), įrašant naudotojo vardą, teises (administratorius ar ne), laiką, vykdomą klasės metodą (klasės pavadinimas + metodo pavadinimas). Tam, kad auditavimo kodą įdėti/pašalinti, neturi reikėti modifikuoti/perkompiliuoti audituojamo sistemos kodo {*Java EE/CDI Interceptors*}.
7. [[Extensibility/Glass-box extensibility](#)] Turi būti įmanoma po sistemos sukūrimo į sistemą pridėti naujus į rezervavimo pirmumo grupių skirstymo algoritmus taip, kad nereikėtų modifikuoti ir kompiliuoti seno kodo (būtų pridodamas tik naujas, ir galimai redaguojamas konfigūracinis failas). {*CDI Alternatives*}.
8. [[Extensibility/Glass-box extensibility](#)] Sistema turi būti organizuota taip, kad po jos sukūrimo sistemą būtų galima papildyti nuolaidų sistemomis (pvz., kažkiek procentų nuolaida šį mėnesį gimusiems, Kalėdų/Velykų laikotarpio nuolaida, ...). Naujų nuolaidų sistemų kodo pridėjimas į sistemą bei įjungimas/išjungimas turi būti atliekamas nekeičiant seno kodo (tik redaguojant kokį nors konfigūracinį failą). {*CDI Decorators*}.