ChatGPT

Speksin päivitykset ja uudet liitteet

Tämä dokumentti kokoaa suositellut toimenpiteet speksin ja toteutuksen yhdenmukaistamiseksi, ei-toiminnallisten vaatimusten lisäämiseksi sekä audit- ja RBAC-mallin, hyväksymiskriteerien ja sanaston dokumentoimiseksi. Sisältö on jäsennelty osioihin, jotka voidaan sisällyttää `spec`-hakemistoon tai linkittää projektiin myöhemmin.

1. Taulujen nimien ja sarakkeiden yhdenmukaistus

Koodissa monet taulut ovat monikkomuotoisia ja sisältävät `tenant_id`-sarakkeen. Speksissä käytettiin pääosin yksikkömuotoa. Suosituksena on päivittää speksi vastaamaan koodia, jotta kehittäjät ja testaus voivat tukea samaa terminologiaa.

Suositellut muutokset

- **Taulujen nimet** – Vaihda speksissä yksikkömuodot monikkoon:
- `plan` → `planning_events`
- `forecast_event` → `forecast_events`
- `forecast_line` → `forecast_event_lines`
- `mapping_version` → `mapping_versions`
- `mapping_line` → `mapping_lines`
- **Sarakkeet** – lisää speksiin `tenant_id` kaikille tauluille sekä tarvittaessa `event_time` suunnittelutapahtumiin ja `created_at` / `updated_at` historiaan. Varmista, että status- ja tyypitys-sarakkeiden nimet vastaavat koodin enum-arvoja.
- **Sanasto** – speksiin kannattaa lisätä sanasto, joka määrittelee termit (suomi ja englanti) ja selventää, että taulujen nimet ovat monikkomuotoisia.

2. Ei-toiminnalliset vaatimukset

Hyvät SaaS-järjestelmät kattavat myös suorituskyky-, turvallisuus- ja luotettavuusvaatimukset. Näiden lisääminen speksiin tukee laadunvarmistusta ja compliance-työtä.

Suorituskyky

- Määritä tavoiteltu vasteaika API-päätepisteille (esim. 95-prosenttipiste alle 300 ms yrityksen sisäisille kutsuille).
- Määritä raportointikyselyiden maksimiaikataulut ja varmista, että suositellut indeksit tukevat tätä.
- Kirjaa budjetoidut resurssit (CPU, muisti) SaaS-ympäristölle ja mitoitus useille asiakkaille (per-tenant kuormitus).

Turvallisuus

- Toteuta **rivitasoinen turvallisuus (Row Level Security, RLS)** PostgreSQL:ssä, jotta kyselyt suodattuvat automaattisesti `tenant_id`-kentän perusteella.

- Varmista, että kaikki pysyvät tallenteet (tietokanta ja tiedostot) ovat salattuja levolla (Disk Encryption) sekä liikenteessä (TLS). Erityisesti `app_audit_log` ja käyttäjätiedot.
- Käytä turvallisia hash-funktioita (esim. bcrypt) kaikille salasanoille ja varmista salasanapolitiikat (minimipituus, monimutkaisuus).
- Dokumentoi GDPR:n ja mahdollisten sertifikaattien (SOC 2, ISO 27001) noudattaminen.

Varmuuskopiointi ja palautus

- Määritä automaattinen varmuuskopiointitahti (esim. päivittäin), varmuuskopioiden säilytysaika (esim. 30 päivää) ja testatut palautusmenettelyt.
- Toteuta per-tenant varmuuskopiointi tai varmista, että palautus voidaan kohdentaa yksittäiseen tenanttiin.
- Kuvailkaa disaster recovery –suunnitelma: mitä tapahtuu vakavan häiriön sattuessa ja kuinka nopeasti palvelu palautetaan (RTO/RPO).

3. Audit-logi ja RBAC-malli

Audit-logi

- **Audit-logien periaate** – kaikki kriittiset toiminnot (kirjautuminen, roolin myöntäminen, suunnitelmien ja ennusteiden luominen, mapping-versioiden julkaisu) kirjataan append-only-auditiin.
- **Sisältö** – jokainen lokimerkintä sisältää vähintään: aikaleima, käyttäjän tunniste, tenantin tunniste, toiminto (kategoria ja kuvaus), kohteen tunniste (esim. projekti, versio) sekä mahdollinen tapahtuman tulos (onnistui/virhe).
- **Säilytyspolitiikka** – määritä audit-logien säilytysaika (esim. 1–5 vuotta riippuen vaatimuksista) ja arkistointiprosessi. Lokit tulee säilyttää tamper-evident-muodossa (esim. WORM-säilytys tai hash-ketjut).
- **Näkyvyys** – RBAC-mallin mukaan vain auditorit tai turvallisuusroolissa toimivat henkilöt voivat lukea kaikkien tenanttien audit-lokeja, muut käyttäjät näkevät vain omiin tenantteihin liittyvät lokit.

RBAC-malli

- **Rakenne** – oikeudet määritellään roolien (esim. `owner`, `admin`, `viewer`) kautta ja roolit ovat aina sidottuja tenanttiin (organization/project). Käyttäjällä voi olla eri rooli eri projektille.
- **Periytyvyys** – roolit periytyvät hierarkiassa (group → organization → project) ja alataso voi vahvistaa tai rajoittaa periytettyä roolia.
- **Kustomointi** – tarjoa mahdollisuus luoda tenant-kohtaisia rooleja, joissa voidaan määrittää sallittavat toiminnot (CRUD useille entiteeteille).
- **Päätöksenteko** – jokainen pyyntö arvioi käyttäjän roolin ja tenantin; luku/kirjoitusoikeus myönnetään, jos rooli sallii toiminnon kyseisessä tenantissa.

4. Hyväksymiskriteerit ja testaus

Lisää speksiin osio, joka määrittelee hyväksymiskriteerit kullekin toiminnallisuudelle. Kriteerien perusteella voidaan laatia testejä (yksikkö-, integraatio- ja E2E-testit).

Esimerkkikriteerit

- **Mappingin versionointi** – kun uusi mapping julkaistaan, vanhan version `ends_at` asetetaan ja uusi versio alkaa automaattisesti; järjestelmä estää ennusteen luomisen, ellei mapping ja suunnitelma ole lukittuja. Hyväksyntä: testaa, että mappingin versiot vaihtuvat oikein ja että ennusteiden luonti hylätään, jos versio ei ole lukittu.
- **Forecast-gating** – ennusteen luominen vaatii, että projekti on olemassa, mapping on lukittu ja suunnitelma on hyväksytty; testaa myös edge-caset (esim. tyhjä mapping).
- **RBAC-valvonta** – testaa, että eri rooleilla (owner, admin, viewer) on oikeat luku-/kirjoitusoikeudet ja että tenant-eristys toteutuu. Automaatiotestit injektoivat erilaisia `tenant_id`-arvoja ja varmistavat, ettei toisen tenantin dataa voi lukea tai muokata.
- **Audit-logien tarkistus** – testaa, että jokaisesta kriittisestä toiminnosta syntyy lokimerkintä, ja että lokit sisältävät vaaditut kentät; testaa myös säilytyksen täyttyminen (esim. logit arkistoidaan).

5. Sanasto ja i18n

Selkeä terminologia vähentää väärinymmärryksiä, erityisesti kun speksissä käytetään sekä suomen- että englanninkielisiä sanoja.

Sanasto

Termi (fi)	Termi (en)	Selitys
suunnitelmatapahtuma	planning event	Tapahtuma, jolla tallennetaan yhden projektin suunnittelun tila tiettynä ajankohtana; tallentaa budjetoidun aikataulun ja oletukset.
ennustetapahtuma	forecast event	Tapahtuma, joka sisältää ennusteen rivit (`forecast_event_lines`) ja liittää ennusteen projektiin ja mapping-versioon.
mapping-versio	mapping version	Versio, joka määrittää kohteen (target) ja työkoodien (work codes) väliset suhteet.
audit-lokimerkintä	audit log entry	Rivi audit-logissa, joka tallentaa käyttäjän ja tenantin tekemän toiminnon aikaleimoineen.
tenant-id	tenant id	Monivuokraisen SaaS-järjestelmän tunniste, jolla erotetaan eri yritysten tiedot toisistaan.

i18n

- Valitse pääasialliseksi dokumentointikieleksi englanti, mutta säilytä tarvittaessa suomenkieliset käännökset. Toteuta sovellukseen i18n-kerros, jossa käyttöliittymäelementit voidaan kääntää.
- Huolehdi, että tietomallin nimet ovat englanniksi ja käyttöliittymässä voidaan näyttää lokalisoidut nimikkeet (esim. “Suunnittelutapahtuma (Planning Event)”).

Soveltaminen: Tämä asiakirja voidaan liittää projektin `spec`-hakemistoon (esim. `spec/05_nonfunctional_and_security.md`). Toteutuksen päivittäminen speksin mukaiseksi edellyttää

lisäksi muutoksia tietokantamigraatioihin ja koodiin, mikä tulee tehdä hallitusti (pull request -prosessin kautta).
