

Varjendid

Lähteülesanne arendushankele

Koostaja: Maa- ja Ruumiamet

DETSEMBER 2025

Sisukord

Sissejuhatus ja taust	3
1. Hanke eesmärk ja ulatus.....	3
2. Tööde ajakava	3
3. Arendustööd	4
3.1. Andmebaasi täienduste tegemine.....	4
3.2. Turvalise andmebaasi loomine	5
3.3. Ehitise kehtivate andmete vaate täiendamine (kasutajaliidese arendus + klassifikaatorite loomine + andmebaasi muudatused).....	9
3.4. Dokumentide vaadete täiendamine	10
3.5. Andmeparanduste tegemine.....	11
3.6. Failide allalaadimise logimise funktsionaalsuse arendamine	11
3.7. X-tee päringu loomine	11
4. Üldised nõuded	12
Lisad.....	12

Sissejuhatus ja taust

E-ehituse platvorm on Maa- ja Ruumiameti poolt hallatav ehitiste ja ehitamisega seotud teenuste ning andmekogude sisendpunkt – avatud keskkond, mis võimaldab erinevate seotud rakenduste kokku toomist, eesmärgiga tagada standardiseeritud ja korrektse info liikumine poolte vahel digitaalselt ning kadudeta ([E-ehitus](#) | [Maa- ja Ruumiamet](#)).

Olulise osa e-ehituse platvormist moodustab ehisregister (EHR), mis on keskne riiklik register, mille abil hallatakse ehitiste infot ning viiakse läbi erinevaid ehitustegevusega seotud menetlusi.

Hädaolukorra seaduse kohaselt on alates 01.07.2026 ehitusluba taotleval isikul kohustus tagada üle 10 000 m² suletud netopinnaga hoonetes, mida külastavad suured rahvahulgad, avalik varjend ning üle 1200 m² suletud netopinnaga elu- ja muudes hoonetes, sh üle 1500 m² tööstus- ja laohoonetes, mitteavalik varjend.

Juhul kui hoone vastab nii mitteavaliku kui ka avaliku varjendi rajamise nõude kriteeriumidele, tuleb rajada avalik varjend. Täpsemad nõuded varjenditele (nt selle suurus, mahutavus jms) kehtestatakse eraldi vabariigi valitsuse määrusega.

Kuna varjendite info on ehitiste kohta käiv andmestik, siis on vaja seda ka menetleda, mistõttu hakatakse seda koguma EHR-i.

1. Hanke eesmärk ja ulatus

Käesoleva hanke eesmärk on varjumise paremaks korraldamiseks luua võimalus eristada e-ehituse platvormil hooneid, milles asub avalik või mitteavalik varjend ning kajastada varjendite teavet kaardil. Lisaks ehitise kehtivatele vaadetele peab see info olema ka taotluste ja teatiste peal, juhul kui ehitises on varjend. Sellest tulenevalt teha platvormil kuvamiseks vajalikud arendustööd, mis on täpsemalt kirjeldatud hankedokumendi peatükis 3.

2. Tööde ajakava

Arendustööde kavandamisel arvestada järgmisega:

- Töö toimub agiilse arendusena sprintide kaupa.
- Enne arendustööde teostamist tuleb teha **detailprojekt** teostatavate tööde planeerimiseks, mis tuleb enne arendustööde algust **kooskõlastada** tellijaga (projektiplaani arvestada ka piisav ajavaru selleks, et tellija jõuaks sellele oma tagasiside anda). Dokumendi sisendiks on käesolev lähteülesanne koos selle lisadega.
- **Eeldatav detailprojekti sisu:**
 - UI Figma vaated + kasutajate interaktsiooni ning süsteemi vastuste täiendamine ja kirjeldamine (läheldes olemasolevast ehitusloa taotlemise protsessist);
 - kontekstidiagramm, kus on näidatud seosed väliste süsteemidega: X-tee ühendus Ohutusportaaliga;
 - seosed äriprotsessi protseduuride ja API teenuste *endpoint*ide vahel;
 - loodavate või muudetavate API-de Openapi kirjeldused;
 - kasutatavate API-de vahelise infovahetuse ja API-de omavaheliste seoste kirjeldused;
 - loodavate või muudetavate UI komponentide nimekiri (kirjeldus);
 - API ja UI komponentide seosed, API-de seosed andmemudelitega;

- UI ja raamkomponendi jagatud *state* kirjeldused, UI kasutatavad *state* kirjeldused;
- loodavate automaattestide nimekiri, mis sisaldab viiteid testitavatele äriprotsessi protseduuridele.
- Arendustööde üleandmise tähtaeg tellijale on hiljemalt **15. juuni 2026 (selleks ajaks peavad olema kõik testimised tehtud ja vead parandatud)**.
- Täpne *live*-tarneaeg lepitakse tellijaga kokku projekti avakoosolekul (tuleneb seadusest tulenevatest tähtaegadest – hiljemalt 1. juuli 2026 peab arendus *lives* olema).

Pakkumuse projektiplaanis ja maksumuse tabelis tuua välja töötundide arv, kulu meeskonnaliikmete kaupa ning hinnastada eraldi kõik järgmised arendustööd, mis on vaja teha seoses varjendite andmete EHR-i lisamisega:

1. turvalise andmebaasi loomine (turvaandmete API);
2. ehitise kehtivate andmete vaate täiendamine (kasutajaliidese arendus + klassifikaatorite loomine + andmebaasi muudatused);
3. dokumentide vaadete täiendamine;
4. ehitise kehtivate andmete ja dokumentide pdf-vormide muudatuste tegemine;
5. detailotsingu juurde varjendite otsingu lisamine (uute TEDI komponentidega);
6. failide allalaadimise logimise funktsionaalsuse arendamine;
7. andmeparandustööde tegemine;
8. X-tee päringu loomine Ohutusportaaliga (ühesuunaline).

Andmeid esialgu ei ole ega kohe teki, vaid kasutajad hakkavad neid süsteemi kandma ehitusloa taotlemise ajal. Arendustööde käigus tuleb lisada vajalikud andmeväljad UI-sse ja andmebaasi.

Baaskomponente käesoleva hanke raames tehtavate arenduste käigus ei muudeta – s.o teostada olemasoleva UI edasiarendus.

3. Arendustööd

3.1. Andmebaasi täienduste tegemine

Varjend on eraldi fiktiivne (eraldatud tunnustega) hooneosa, mis on seotud ehitisega ning mis lisaks ID-le saab ka osakoodi. Fiktiivsesse hooneosasse võib mahtuda mitu hooneosa ja mitu fiktiivset osa võib olla ka ühes hooneosas (st ei pea tegema mitu mitmele suhet), millel on küljes tunnus, et tegemist on varjendiga (kasutajaliidises seda ei näe).

Andmebaasi teha järgmised täiendused:

- Teha juurde üks hooneosa tüüp – varjend (turvabaasi tabelis `do_ehitis_osa.liik = 'V'`).
- Hooneosadele panna külge tehnoandmed, st tuleb täita `do_eh_tehna` tabel järgmise punkti klassifikaatoritega ning panna viidad ehitisele ja varjendi osale.
- Tehnoandmete tabelisse lisada **klassifikaatorid**, mida hakkavad varjendid kasutama (tabelisse `kl_tehnilised_naitajad`):

- varjendi liik (avalik/mitteavalik);
- rajamise aasta;
- varjumispind;
- kaitseklass.
- Varjendi baasi panna juurde **ehitise andmed**: ehitis, ehitise osa ja tehnoandmed.

3.2. Turvalise andmebaasi loomine

Turvalise andmebaasi loomise nõuded

1. ÜLDINE KIRJELDUS

1.1 Eesmärk

Lua eraldiseisev turvaline andmebaas hoonete tehniliste andmete turvaliseks säilitamiseks ja haldamiseks, mis tagab andmete piiratud juurdepääsu ning säilitab seosed olemasoleva EHITUS süsteemi hoonete versioonidega.

1.2 Olemasolev süsteem

- **Primaarne andmebaas:** EHRDB
- **Hoonete tabel:** eehitus.do_ehitis (unikaalne ID: do_ehitis_id)
- **Tehniliste andmete tabel:** do_eh_tehna (võti-väärtus formaat)
- **Klassifikaatorite tabelid:**
 - koodid.kl_klassifik_tyyp (klassifikaatori tüübid – vajab täiendamist)
 - koodid.kl_tehnilised_naitajad (tehniliste andmete väärtused)
- **Seotud objektid:**
 - **Kehand:** do_ehitis_kehand (ID: do_ehitis_kehand_id)
 - **Hooneosa:** do_ehitis_osa (ID: do_ehitis_osa_id)
- **Tehniliste andmete seosed:** kõik objektid (hoone, kehand, osa) võivad omada tehnilisi andmeid do_eh_tehna tabelis

1.3 Projekti ulatus

- Eraldiseisva turvalise andmebaasi loomine piiratud juurdepääsuga tehniliste andmete jaoks.
- Klassifikaatori tabeli koodid.kl_klassifik_tyyp täiendamine turvalise märgiga.
- Automaatse andmete eraldamise süsteemi väljatöötamine klassifikaatorite alusel.
- Mikroteenuste arhitektuuri väljatöötamine.
- Andmete sünkroniseerimise teenuste loomine.
- Varjendite otsingu lisamine.
- Hoonete versioonihalduse süsteemi täiendamine.

2. FUNKTSIONAALSED NÕUDED

2.1 Andmete eraldamine

2.1.1 Skeemide jaotus

- **Avalik skeem (eehitus):** vabalt kättesaadav informatsioon (olemasolev skeem, midagi juurde ei tee).
- **Turvaline skeem (piiratud_eehitus):** piiratud juurdepääsuga (tehnilised) andmed.

2.1.2 Tehniliste andmete eraldamise kriteeriumid

- **Klassifikaatorite struktuur:**
 - koodid.kl_klassifik_tyyp: klassifikaatori tüübid (tuleb täiendada)
 - koodid.kl_tehnilised_naitajad: tehniliste andmete väärtused
- **Automaatse eraldamise süsteem:**
 - Tabelisse koodid.kl_klassifik_tyyp lisatakse uus veerg on_turvaline (BOOLEAN).
 - Kui klassifikaatoril on märgitud on_turvaline = TRUE, siis kõik selle klassifikaatori alusel sisestatud tehnilised andmed suunatakse automaatselt skeemi piiratud_eehitus.
 - Uute turvaliste klassifikaatorite lisamine toimub projekti käigus.
 - **Turvaline skeem asub samas andmebaasis, kuid eraldi skeemis.**

2.1.3 Andmete töötlemise mudel

- **Document API keskne koordineerimine:**
 - Document API võtab vastu kõik ehitise andmed (avalikud ja turvalised).
 - Identifitseerib turvalised andmed klassifikaatori on_turvaline flag põhjal.
 - Salvestab avalikud andmed kohe eehitus skeemi.
 - Teeb sünkroonse HTTP kutse Secure API-le turvaliste andmete salvestamiseks.
 - Kasutab andmete salvestuseks do_ehitis_id-d.
 - Tagastab unified vastuse kasutajaliidesele.
- **Andmete sünkroniseerimine:**
 - Sünkroonne töötus (kõik sammud järjestikku).
 - Transactional rollback ebaõnnestumise korral.
 - Atomic operation: kõik andmed salvestatakse või mitte midagi.

2.2 Andmebaasi arhitektuur

2.2.1 Turvaline andmebaas

- **Nimi:** piiratud_eehitus (skeemi nimi)
- **Asukoht:** sama EHRDB andmebaasis eraldiseisvas skeemis
- **Turvalisus:** skeemipõhine eraldatus koos tugevdatud juurdepääsukontrolliga
- **Varundamine:** varukoopiaid koos põhiandmebaasiga
- **Andmebaasi (skeemi) juurdepääs:** luuakse eraldi kasutaja

2.2.2 Põhitabelite struktuur

- **Turvaline skeem sisaldab:**

- **Tabelite struktuur:** sarnane eehitus baasiga, luuakse ainult vajalikud tabelid
- **Peavõtmete väärtused:** do_ehitis_id etc langevad kokku põhiandmebaasiga
- Luuakse luba_loend väli vastavuses 'Juurdepääsupiirangud' dokumendiga
- **ID sünkroniseerimine:**
 - Mõlemas skeemis kasutatakse samu primary key väärtusi
 - do_ehitis_id, do_ehitis_osa_id, do_eh_tehna_id on identsed
 - Lihtne cross-schema päring – sama ID-ga kirjed kuuluvad kokku

2.2.3 Skeemide eraldamise põhimõtted

KEELATUD cross-schema operatsioonid:

- ✗ JOIN-id skeemide vahel
- ✗ Foreign key constraints skeemide vahel
- ✗ Shared stored procedures
- ✗ Cross-schema views
- ✗ Direct SQL access teisest skeemist

LUBATUD operatsioonid:

- ✓ API-põhine kommunikatsioon
- ✓ HTTP calls teenuste vahel
- ✓ Sama ID väärtuste kasutamine
- ✓ Independent transactions
- ✓ Separate connection pools

Andmete jagamine: ainult API kaudu.

2.3 Teenuste arhitektuur

Turvalise andmebaasi jaoks luuakse juurdepääsupiiranguga teenused. Kõik loodavad teenused peavad kasutama e-ehituse autoriseerimise põhimõtteid, mis on toodud MFN-is.

2.3.1 Teenuste loetelu

- **Document API muudatused:**
 - Võtab vastu kõik ehitise andmed (avalikud + turvalised).
 - Identifitseerib turvalised andmed klassifikaatori flag'i põhjal.
 - Teeb sünkroonse HTTP kutse Secure API-le sama ID-ga.
 - Tagastab unified response'i UI-le.
 - Pole otsest ligipääsu piiratud_eehitus skeemile.
- **Secure API teenused:**
 - POST /secure-data/{buildingId} (kasutab sama ID-d)
 - GET /secure-data/{buildingId} (sama ID põhjal)
 - PUT /secure-data/{buildingId} (sama ID põhjal)
 - DELETE /secure-data/{buildingId} (sama ID põhjal)
 - Kasutab Document API poolt edastatud ID-sid
 - Ainult turvaliste andmete otsene ligipääs

- Pole otsest ligipääsu eehitus skeemile
- Independent database connection pool
- Eraldi authentication ja authorization
- Separate deployment unit
- **Deepcopy ja deepupdate muudatused:**
 - Document API koordineerib mõlema skeemi andmete kopeerimist
 - Secure API kutsed turvaliste andmete versioonide loomiseks
 - Cross-reference uuendamine uute versioonide vahel
- **Turvaliste tehnoandmete otsimine:**
 - Secure API pakub otsingufunktsionaalsust
 - Juurdepääsukontroll vastavalt kasutaja õigustele
 - Filtreerimine ja paginatsioon

2.3.2 API integratsioon ja error handling

- **HTTP kommunikatsioon:**
 - Document API → Secure API: HTTPS
 - Timeout: 30 sekundit
 - Retry: 2 korda, 5 sekundi intervalliga
 - Connection pooling: max 10 ühendust
- **Error handling:**
 - HTTP timeout → rollback transaction
 - Secure API unavailable → rollback transaction
 - Validation errors → tagasta kõik errors UI-le
 - Network errors → retry ja siis rollback
- **Circuit breaker (optional):**
 - 5 järjestikust ebaõnnestumist → open circuit
 - 60 sekundit recovery time
 - Fallback: "Secure API temporarily unavailable"
- **Monitoring:**
 - HTTP request/response logging
 - Error rate tracking (target: <5%)
 - Response time metrics (target: <5 sek)
 - Secure API availability monitoring

2.4 Kasutajaliidese muudatused

2.4.1 Andmete sisestamine

- UI kogub kõik andmed (avalikud + turvalised) ühes vormis.

- Teeb ühe API kutse Document API-le.
- Loading state kuni vastus saabub.
- Error handling: kuvab kõik validation ja system errors.

2.4.2 Turvaliste andmete kuvamine

- Varjendite plokk kuvatakse ainult sisse loginud kasutajatele.
- Juurdepääsukontroll vastavalt kasutaja rollidele.
- **Dokumentide ja ehitise andmete vaatel on varjendite plokk selles olevate avalike ja mitteavalike varjendite andmetega nähtav kõigile sisse loginud kasutajatele.**
- Otsingutes on avalikud varjendid nähtavad kõigile sisse loginud kasutajatele, mitteavalikud varjendid nähtavad ainult KOV-idele, TTJA-le ja Päästeametile (lisaks ka EHR-i kasutajatoele).
 - Samasse kohta kasutajaliideses, kus on detailotsing, lisada juurde uus otsing 'Varjend', mis on nähtav vaid autoriseeritud kasutajale, ja mis otsib turvalisest andmebaasist välja ehitised, milles on varjend (**tabelina**). Autoriseeritud kasutaja saab otsida hooneid, kus on avalik varjend. Mitteavalike varjenditega ehitisi saab otsida kasutaja, kes on TTJA, Päästeameti või KOV kasutaja (lisaks ka EHR kasutajatugi).
 - Otsingu sisu on ehitiste otsimine selle järgi, kas ehitises asub (mitte)avalik varjend. Filtrisse saab panna linnukese [v] ehitised, kus on avalik varjend ja [v] ehitised, kus on mitteavalik varjend. Sealjuures teine linnuke on näha vaid kindlatele kasutajatele. Kui ehitisel on mõlemad varjendid, siis ehitis tuleb ikka nimekirja, aga tabelist ei näe, kas seal on ka mitteavalik varjend, kuna selle järgi ei saa filtreerida. Kui ehitis lahti klikata, siis näeb, kas tal on ainult avalik või on ka mõni mitteavalik varjend.
 - **Otsingu tulemused kuvatakse ka otsingu juures oleva EHR-i kaardile** (lähtudes PLANIS arenduse raames välja töötatud lahendusest – live jaan.2026).

2.4.3 Error handling UI-s

- Validation errors: field-level error messages
- System errors: üldine error message + retry option
- Timeout errors: "Processing failed, please try again"
- Network errors: "Connection problem, check your internet"

3.3. Ehitise kehtivate andmete vaate täiendamine (kasutajaliidese arendus + klassifikaatorite loomine + andmebaasi muudatused)

Kasutajaliidese muudatused

Täiendada olemasolevat ehitise **kehtivate andmete vaadet** lisades **eraldi ploki 'Varjendid'**, mis annab infot selle kohta, kas hoones on varjend (avalik/mitteavalik), mitu tükki neid, millal rajatud on etc. UI-s võib kujundada ka tabelina, mis sisaldab järgmisi andmeväljasid (iga varjendi kohta eraldi andmeplokk):

- varjendi liik – klassifikaatorid: avalik varjend / mitteavalik varjend;
- rajamise aasta;
- varjumispind (m²);
- inimeste arv – süsteem arvutab välja ja täidab selle pinna järgi (valemi x järgi);

- varjendi projekteeritud kaitseklass (V1, V2 vm klass – täpsustub).

Vastavad väljad tuleb lisada nii ehitise andmete vaatele kasutajaliideses kui ka ehitise andmete vaate pdf-versioonile.

NB! 'Varjendid' plokk avaldub vaid sisse loginud kasutajale pärast seda, kui ta on klikanud selle ploki lahti, misjärel pärib turvaandmete API ploki andmed ning ühtlasi logitakse ka nende andmete pärimine.

Sellesse tabelisse, kus täna kasutajaliides on hooneosade loetelu, ei tohi tulla sisse varjendid ja varjendite loetellu peaksid tulema sisse vaid varjendid. Lisaks, igal pool, kus pindalad kokku arvestatakse, ei tohi varjendite pindalasid sisse arvutada, kuna tegemist pole sama pindalaga – UUS pindala klassifikaator varjumispinna jaoks.

Täiendada detailotsingute kasutajavaade uue otsinguga 'Varjend'. Detailotsingute vaatele lisatav varjendite otsing (peab olema eraldi vorm, kuna kasutab turvalist API-t) tuleb teostada **uute TEDI komponentidega**, lisades otsingule oma vahelehe ja filtri. Filtri väljad täpsustada detailanalüüsi käigus.

3.4. Dokumentide vaadete täiendamine

Täiendada olemasolevate dokumentide rakenduse vaateid nii, et varjendite info oleks nähtav ka menetluskeskkonna kaudu esitatavatel taotlustel, teatistel ja haldusaktidel ning menetluse ja toimingute vaates. Täiendada tuleb järgmiste dotyde vaateid:

- 11201 Ehitusteatis;
- 11271 Ehitusloa taotlus;
- 11301 Kasutusteatis;
- 11371 Kasutusloa taotlus;
- 11525 Andmete esitamise teatis;
- 12271 Ehitusluba;
- 12371 Kasutusluba;
- 12994 Haldusakt;
- 12995 Haldusakt (keelav);
- 12996 Haldusakt (lubav);
- 12899 Ettekirjutus;
- 11002 Projekteerimistingimuste taotlus;
- 11802 Projekteerimistingimused ning vastavad keelduvad otsused.

Dokumendi vaatele tuleb lisada 'Varjendid' plokk, mis annab infot selle kohta, kas ehitisse on plaanis rajada/olemas avalik või mitteavalik varjend. Iga varjendi kohta on eraldi andmeplokk (analoogselt nagu hooneosade tabelis saab lisada täiendavaid hooneosasid). UI-s võib kujundada ka tabelina, mis sisaldab järgmisi andmeväljasid:

- varjendi liik – klassifikaatorid: avalik varjend/mitteavalik varjend;
- rajamise aasta;
- varjumispind (m²);
- inimeste arv – süsteem arvutab välja ja täidab pinna järgi (valemi x järgi);
- varjendi projekteeritud kaitseklass (V1, V2 vm klass – täpsustub).

‘Varjendid’ plokk avaldub alles sisse loginud kasutajale.

Teostada vastavad kontrollid ja veateated ehitise andmete täitmisel dokumendil. Kontrollide rakendamise täpsemad nõuded formuleeritakse detailanalüüsi raames.

Mõned üldised põhimõtted, mida kasutaja liideses vormidel vajalike andmete kogumisel arvestada:

1. Andmete sisestamisel eelistada klassifikaator valikuid – kasutaja valib väärtused eeldefineeritud listist.
2. Välja *tooltipis* või abiinfo plokis selgitus ootuste osas, mida kasutaja peab andmete sisestamisel arvestama.
3. Numbriliste väärtuste sisestamisel süsteem kontrollib, et tähemärke ei oleks esitatud.
4. Teatud numbriliste väärtuste korral võib süsteem kontrollida loogilisust alam- ja piirmäära järgi.
5. Numbriliste väärtuste esitamisel võimaldab süsteem esitada eeldefineeritud pikkusega numbrit.
6. Kui esitav number on väiksem (nt NNN), kui eeldefineeritud esitatava numbrilise pikkus (NNNNN), siis süsteem lisab numbrilise kohatäiteks 0-d. Nt kasutaja oodatav numbrilise pikkus on NNNNN, kasutaja saab sisestada NNN – süsteem annab tulemuseks 00NNN.
7. Süsteem ei võimalda andmeid või dokumenti esitada, kui kohustuslikuks märgitud väljad ei ole täidetud. Veateated ja hoiatused kuvatakse kasutajale nagu olemasolevas funktsionaalsuses vormi päises olevates ‘värvilistest’ kastikes (vt Kontrollid ja veateated - e-ehituse platvorm - Confluence).

3.5. Andmeparanduste tegemine

Kõik need hooned ja hooneosad, mille nimetuses on hetkel sees sõna või sõnaosa ‘varjend’, tuleb ära muuta. Asendussõna (või sõnad) leppida kokku detailanalüüsi käigus – pärast seda, kui ehitise nimede väljavõtted on tehtud.

3.6. Failide allalaadimise logimise funktsionaalsuse arendamine

Lisada *file download* teenuse otspunktidele: https://swaggerui.ehr.ee/file_upload_teenus#/common-file-upload-controller/downloadFile_1 ja https://swaggerui.ehr.ee/file_upload_teenus#/file-upload-controller/downloadFile uus funktsionaalsus, mis salvestab andmebaasi do_file_log (uus tabel) **info faili alla laadinud kasutaja kohta**:

- 1) kasutaja ees ja perenime;
- 2) asutuse, mille kasutaja isik oli sel hetkel;
- 3) ip aadressi + muud väljad, mis identifitseerivad faili ja faili allalaadimise fakti.

3.7. X-tee päringu loomine

Lua X-tee päring Ohutusportaaliga (ühesuunaline), mis väljastab kehtivat varjendite infot, kasutades e-ehitus X-tee andmete baasi. Selle realiseerimine eeldab aktualiseerija ja kehtivate andmete rakenduste täiendamist. Hetkel on kasutuses kõige uuem X-tee versioon.

4. Üldised nõuded

Järgnevalt on esitatud üldised nõuded tehtavatele töödele:

1. Tehtud tööd peavad vastama käesoleva lähteülesande põhiosas ning lisades toodud nõuetele.
2. Kui tellija teekides on olemas UI komponendid või mikroteenused, mida saab otse või mõistliku lisatööga käesoleva projekti ülesannete täitmiseks kasutada, kasutab töövõtja neid (mitte ei arenda vastavaid komponente/mikroteenuseid uuesti või kasuta enda vastavaid teeke) – väljaarvatud juhul, kui selles ei ole tellijaga eelnevalt kirjalikult taasesitatavas vormis teisiti kokkulepitud. Enne iga komponendi realiseerimist kooskõlastab töövõtja selle tellijaga.
3. Kasutajavaadete muudatused UI-s tuleb prototüüpida Maa- ja Ruumiameti Figma keskkonnas.
4. Kasutajaliidese arendust teostatakse arenduse käigus koostöös tellijaga.
5. Tööd toimuvad tihedas koostöös töövõtja ja tellija vahel. Töövõtja pakub töö käigus jooksvalt välja omapoolseid lahendusi töö teostamiseks ning arutab need läbi töö tellijaga.
6. Töö lõpptulemused dokumenteeritakse Confluence´s vastavalt Tellija poolt ettenähtud struktuurile ja tarkvaraarenduse heale tavale.
7. Töövõtja peab olema valmis tellija soovil asendama tööloike samaväärsetega, kui töö teostamise käigus selgub, et vahetus aitab paremini saavutada töö eesmärke.

8. Secure-api jõudluse nõuded:

- a. Response time:
 - Ehitise andmete salvestamine: <5 sekundit (95th protsendil)
 - Ehitise andmete lugemine: <1 sekundit
 - Secure API kutse: <30 sekundit (timeout)
- b. Availability:
 - Document API: 99.5% uptime
 - Secure API: 99.0% uptime (vähem kriitiline)
 - End-to-end success rate: >95%
- c. Scalability:
 - Concurrent users: 100 kasutajat
 - Peak load: 500 requests/hour
 - Database connections: max 50 per service

Lisad

Lisa 1 E-ehituse platvormi tehniline arhitektuur: Juhendid arendajale: [Ehitisregister](#)

Lisa 2 [E-ehituse mittefunktsionaalsed nõuded](#)

Lisa 3 [E-ehituse platvormi arendustööde läbiviimise ja testimise juhend](#)

Lisa 4 [E-ehituse platvormi stiiliraamat](#)

Lisa 5 [Kasutajaõigused e-ehituse süsteemis](#)

Lisa 6 [Kontrollid ja veateated - e-ehituse platvorm](#)