**Tehniline kirjeldus - Toetuste ja üldkomponentide II etapi arendused**

**Lisa 1 - Tehniline kirjeldus**

# Töö sisu ja saavutatav tulem

* 1. Sotsiaaltoetuste ja -teenuste andmeregistri (STAR1 ja STAR2) toetuste ja seotud üldkomponentide arendamine ja analüüs, vigade tuvastamine, parandamine, testimine ning tarkvara tarnimine tellija keskkonda;
  2. Saavutatavad tulemid: Sotsiaaltoetuste ja -teenuste andmeregistri (STAR1 ja STAR2) toetuste ja seotud üldkomponentide analüüs, arendus- ja muud seotud tööd ning vigade tuvastamine ja parandamine;
  3. Saavutatud kokkulepped: toetuste ja seotud üldkomponentide arendus ja analüüsitööde täpsustatud projektiplaan ning skoop lepitakse kolme osapoole (täitja, TEHIK ja SKA) vahel kokku pärast arendaja poolt teostatud eelanalüüsi viivitamatult. Kokkulepitud projektiplaan saab aluseks tööde teostamisele. Projektiplaanis on välja toodud etapid. Viimaste sisu ja tähtajad lepitakse poolte vahel kokku lepingu täitmisel. Projektiplaan ja skoop võib ühiste eesmärkide täitmise nimel SKA ja TEHIK’u kirjalikul tellimusel ajas muutuda, kuid muudatuseni on see pooltele siduv dokument, millest tuleb tööde teostamisel kinni pidada kuni lepingu lõpuni või muudatuse tegemiseni. Tegemis on RHS § 123 lg 1 p 2 muudatustega.

# Projekti eesmärk

* 1. Projekti II etapi eesmärk on teostada analüüs ja luua võimekus toetuste menetlemiseks STAR2-s. Kuna toetused, mida tuntakse ka lihttoetustena (universaaltoetustena), defineeritakse üle Eestiliselt erinevalt, luuakse tingimused selleks, et iga kohalik omavalitsus saaks seadistada ja menetleda toetusi infosüsteemi vahendusel vastavalt kehtestatud korrale ja tingimustele.

# Projekti tulemid

* 1. II etapi raames võetakse fookusesse valitud toetuste üle toomine uuele platvormile.
  2. II etapi tulemusena on loodud võimekus:
     1. toetuste haldamiseks;
     2. toetuste menetlemiseks ja haldusotsuste koostamiseks vajaliku täiendava lisafunktsonaalsuse loomine;
     3. toetuste väljamaksete ja aruandluse lahenduse täiendamine;
     4. kõik muud olulised lisa funktsionaalsused kasutusmugava lahenduse väljatöötamiseks;
     5. Antud projekti raames teostatakse ka STARi üldkomponentidega seotud vajalikud tööd.

# Tööprotsess

* 1. Järgnevalt täpsustatakse raamlepingus kirjeldatud tingimusi tööprotsessi, dokumenteerimise, töökorralduse, testimise ja mittefunktsionaalsete nõuete osas.
  2. **Töökorraldus**
     1. Projekti töökorraldus toetub kodukorrale;
     2. Täpsem tööprotsess lepitakse poolte vahel kokku lepingu täitmisel;
     3. Tellija ei ole kohustatud vastu võtma tehtud arendustööde tarneid, kui arendustööl puuduvad automaattestid;
     4. Arendustööd kaetakse tellija arhitektiga kokku lepitud ulatuses feature flag’dega.
  3. **Tööde üleandmine**
     1. Arendustööde lähtekoodi tarne toimub pideva integreerimise (continuous integration) teel;
     2. Tööde järjekord, tarneprotsess, üleandmise aeg ja UAT tagasiside andmise aeg lepitakse kokku tööde käigus taasesitamist võimaldavas vormis (Tellija ja Täitja poolt);
     3. Tellija korraldab tulemite üleandmise järel vajalikud lähtekoodi paigaldused keskkondadesse;
     4. Täitja viib läbi testkeskkonnas suitsutesti;
     5. Tellija viib läbi vastuvõtutestimise (Acceptance Test, User Acceptance Test);
     6. Täitja teab, et tellija võib igat üle antud funktsionaalsust paigaldada toodangusse ning täitja peab tagama, et lõplikult valmimata funktsiooni protsessid on võimalik toodangust välja lülitada (feature flags). Funktsioonide sisse-välja lülitamise võimekus lepitakse Tellija ja Täitja vahel kokku tööde käigus.
  4. **Dokumentatsioon**
     1. Vastavalt alapunktis 4 kirjeldatud arendustööde skoobile tuleb töö käigus luua ja täiendada STARi dokumentatsiooni, milleks on minimaalselt:
        1. Funktsionaalsuse ja äriprotsesside kirjeldused kokkulepitud kujul (nt BPMN standard, diagrammid ja kirjeldused Confluence’is) koos ärireeglitega, mida peab olema hiljem võimalik täiendada järgnevates etappides lisanduva funktsionaalsuse alusel ning siduda tervikliku protsessi- ja ärireeglite kirjeldusega;
        2. Tehniline dokumentatsioon (nt paigaldusjuhendid Confluence keskkonnas);
        3. Andmemudel;
        4. Kasutusjuhend;
     2. Dokumentatsiooni täpsed liigid ja sisu formaat lepitakse kokku tööde tellimise käigus.
  5. **Nõuded automaattestimisele**
     1. Loodav lähtekood peab olema kaetud 75% ulatuses ühiktestidega, 50% ulatuses integratsiooni/liideste/süsteemi/”end-to-end” automatiseeritud testidega;
     2. Unit testid käivitatakse enne koodi kokku ehitamist;
     3. Unit testid on üks osa üleantavast koodist;
     4. Testidega kaetakse serveripoolne ja kasutajaliidese kood, kui arendaja ei lepi TEHIKu arhitektiga kokku teisiti;
     5. Funktsionalsed automaattestide tegemine lepitakse Tellija Täitja poolt kokku tööde käigus;
     6. Tellija ja täitja peavad koostöös panema arenduskeskkonnas olevad automaattestid (API testid) tööle ka vähemalt testkeskkonnas või kui on kokkulepitud, siis ka teistes olemasolevates keskkondades;
     7. Automatiseeritud testiliikide haldamisega ja ajakohasena hoidmisega tegeleb arendaja;
     8. Automaattestide raporteerimiseks seatakse TEHIKu koodihoidla keskkonnas (Gitlab) üles pideva integratsiooni töövoog, mis seadistatakse SonarQube koodi ja testide kaetuse analüsaatoriga koos toimima;
     9. SonarQube's kontrollitavate vigade ja haavatavuse taseme kontroll peab olema edukalt läbitud;
     10. Kõik testide automatiseerimise tööriistad/raamistikud peavad oskama töötada GitLab PipeLine's ja Docker Container's;
     11. Liideste testimisel tuleb lisaks positiivsele töövoole katta ka negatiivseid töövooge;
     12. Liideste testide lähtekood peab olema kättesaadav GitLabis;
     13. Automaattestid peavad üle antavas lähtekoodis pideva integratsiooni töövoo edukalt läbima;
     14. Arendaja peab hoolt kandma, et varasemalt loodud testid jäävad töö lõpetamisel edukalt läbitavaks;
     15. Nõuete muutumisel peab arendaja viima testid vastavusse uute nõuetega või neid eemaldama, kui selliseid nõudeid enam ei eksisteeri;
     16. Mõistlik ulatus kasutajaliidese testimisest peab olema automatiseeritud ja korduvkasutatav tööde käigus kokkulepitud raamistikul;
     17. Automaattestimise (API test) artefaktidena luuakse:
         1. Paigaldusjuhend automaattestide integreerimiseks testkeskkonda ja teistesse kokkulepitud keskkondadesse;
         2. Kasutusjuhend automaattestide käivitamiseks.
  6. **Koormustestimised**
     1. Rakendus peab toetama paralleelselt 500 päringut (siinkohal on oodatav arendajapoolne hinnang, kas see on piisav päringute maht, vajadusel tõsta arvu);
     2. API teenustele ning keskmine päringu teenindamise aeg ei tohi olla suurem kui 3 sekundit (peab hindama API päringu keerukust). Projekti käigus täpsustatakse vastamiseajad;
     3. Nõude täitmise kontrolliks tuleb luua koormustestid. Koormustestid peavad olema juhitavad muutujatega ehk seadistatavad. Koormustestide tegemine lepitakse tellija ja täitja poolt kokku tööde käigus.
  7. **Mittefunktsionaalsed nõuded**
     1. Mittefunktsionaalsete nõuete täitmisel tuleb lähtuda järgnevast:
        1. Uue funktsionaalsuse loomisel tuleb lähtuda raamlepingus ja selle lisades kirjeldatud nõuetest;
        2. Täpsemad mittefunktsionaalsed nõuded, millest lähtuda, asuvad Confluence keskkonnas: <https://tehik.ee/arendusjuhendid> ja <https://wiki.sm.ee/pages/viewpage.action?pageId=42184818>;
        3. Põhjendatud juhtudel võib seatud nõuetest kõrvale kalduda taasesitamist võimaldavas vormis sõlmitud kokkuleppel tellijaga.