



## Lisa 1 - Tehniline kirjeldus

### 1. Töö sisu ja saavutatav tulem

- 1.1. Töö sisu: Sotsiaaltoetuste ja -teenuste andmeregistri (STAR<sub>1</sub> ja STAR<sub>2</sub>) toetuste ja seotud üldkomponentide arendamine ja analüüs, vigade tuvastamine, parandamine, testimine ning tarkvara tarnimine tellija keskkonda;
- 1.2. Saavutatavad tulemid: Sotsiaaltoetuste ja -teenuste andmeregistri (STAR<sub>1</sub> ja STAR<sub>2</sub>) toetuste ja seotud üldkomponentide analüüs, arendus- ja muud seotud tööd ning vigade tuvastamine ja parandamine;
- 1.3. Saavutatud kokkulepped: arendus ja analüüsitööde täpsustatud projektiplaan ning skoop lepatakse 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, mille sisu ja tähtajad lepatakse poolte vahel kokku lepingu täitmisel. Projektiplaan ja skoop võib ühiste eesmärkide täitmise nimel SKA, TEHIK'u ja täitja kirjalikul tellimusel ajas muutuda, kuid muudatuseni on see pooltele siduv dokument, millest tuleb tööde teostamisel kinni pidada kuni lepingu lõpuni või muudatuse tegemiseni.
- 1.4. Tööde teostamise tähtajad on märgitud lepingus.

### 2. Projekti eesmärk

Projekti eesmärk on teostada analüüs ja luua võimekus toimetulekutoetuse menetlemiseks STAR<sub>2</sub>-s. Lisaks täiendada juba arendatud toetuste menetluste töövooge selliselt, et oleks võimalik viia STAR<sub>1</sub>s menetlemise vajadus võimalikult madalale.

### 3. Projekti tulemid

- 3.1. Projekti tulemusena on võimalik määrata toimetulekutoetust STAR<sub>2</sub>-s, selleks on:
  - 3.1.1. analüüsitud ja arendatud vajalike dokumentide sisestamine ja nende haldamine STAR<sub>2</sub>-s;
  - 3.1.2. analüüsitud ja arendatud perekonna moodustamine, perekonnas osalemine ning sotsiaalne seisund STAR<sub>2</sub>-s;
  - 3.1.3. analüüsitud ja arendatud x-tee päringuid ning nende rakendamist menetluse läbiviimiseks;
  - 3.1.4. välja töötatud ja arendatud isikute toimetulekupiiri arvestus, sissetulekute ja kulude arvestuste loogika ning summade kalkuleerimiste süsteem STAR<sub>2</sub>-s;
  - 3.1.5. analüüsitud ja arendatud kõikide erandite rakendamised ja muud toimingud, mis vajalikud, et toimetulekutoetust saaks menetleda ainult STAR<sub>2</sub>-s.
  - 3.1.6. välja töötatud lahendus, kuidas KOV-id liiguvad sujuvalt STAR<sub>1</sub>-st STAR<sub>2</sub> menetlema.
- 3.2. Lisaks peavad valmima toetuste jätkutööd, et saaks sulgeda toetuste menetlemine STAR<sub>2</sub>-s. Selleks tuleb:
  - 3.2.1. täiendada menetluse funktsionaalsust, et saaks:
    - 3.2.1.1. menetleda toetust surnult sündinud lapsele, kellel ei ole isikukoodi;
    - 3.2.1.2. menetleda toetust isikule, kellel on välismaa isikuood;
    - 3.2.1.3. menetleda toetust, kui taotlejaks on KOV.
  - 3.2.2. luua andmevahetus Töötukassaga, et edastada infot, et isik on hooldaja;
  - 3.2.3. luua sidusus hooldaja seadmisega STAR<sub>2</sub>-s (arendus valmib teise projekti alt). Üle kontrollida valminud arendust ja täiendada vastavalt vajadusele;
  - 3.2.4. luua menetluse võimalus, kus taotleja erineb isikust- kui on eestkostja/eestkostetav, siis otsus adresseeritud taotlejale;



- 3.2.5. täiendada otsuse allkirjastaja valikut (teises projektis valminud);
- 3.2.6. täiendada toetuste seadistuste osa vastavalt vajadusele (näiteks: lisada võimalus määrata perioodilist toetust 1 kord 12 kuu jooksul)
- 3.2.7. kontrollida ESD deklaratsiooni esitamise funktsionaalsust ning vajadusel täiendada;
- 3.2.8. STST
- 3.2.9. teha muid täiendusi, TEHIKuga kokkuleppel, et liikuda toetuste menetlemise sulgemise poole STAR1-s.

#### 4. Nõuded meeskonnale

Projekti meeskond **peab** koosnema **kahest tiimist**, kes teevad töid paralleelselt. Suurem arenduse vajadus on p-s 3.1 välja toodud meeskonnal, kes peab terve toimetulekutoetuse arenduse valmis saama käesoleva projekti raames.

Teine meeskond vastutab p-s 3.2 tööde eest. Mõlemad tiimid peavad omavahel tegema tihedat koostööd. Arendustel on kattuvaid teemasid.

**Soovitavad meeskonnad** (välja toodud rollid on tellija poolt soovituslikud, kui pakkuja oskab projekti eesmärkide täitmisel põhjendada nimetatud rollidele täiendusi või ühendada mõnda rolli on see, koostöös tellijaga, võimalik:

Roll 3.1 tööde juures	Koormus
projektijuht/tiimi juht	0,5
scrum master	0,5
FE analüütik	1.0
BE analüütik	1.0
BE arendaja	1.0
BE arendaja	1.0
FE arendaja	1.0
testija	1.0
UI/UX	0.75

Roll 3.2 tööde juures	Koormus
projektijuht/tiimi juht	0,2
scrum master	0,2
analüütik (FE ja BE)	1.0
arendaja (FE ja BE)	1.0
testija	0,5
UI/UX	0.25

#### 5. Tööprotsess

Järgnevalt täpsustatakse raamlepingus kirjeldatud tingimusi tööprotsessi, dokumenteerimise, töökorralduse, testimise ja mittefunktsionaalsete nõuete osas.

##### 5.1. Töökorraldus

- 5.1.1. Projekti töökorraldus toetub kodukorrale;
- 5.1.2. Täpsem tööprotsess lepatakse poolte vahel kokku lepingu täitmisel;
- 5.1.3. Arendused võetakse vastu neljas etapis. Iga etapi lõpus esitab täitja tellijale tööde üleandmise akti. Etappide sisu ja kestvused lepatakse kokku kolmepoolselt arenduste alguses, Arveldamine toimub igakuiselt aktide alusel.



- 5.1.4. Tellija ei ole kohustatud vastu võtma tehtud arendustööde tarneid, kui arendustööl puuduvad automaattestid;
- 5.1.5. Arendustööd kaetakse tellija arhitektiga kokku lepitud ulatuses *feature flag*'dega.

## 5.2. Tööde üleandmine

- 5.2.1. Arendustööde lähtekoodi tarne toimub pideva integreerimise (continuous integration) teel;
- 5.2.2. Tööde järjekord, tarneprotsess, üleandmise aeg ja UAT tagasiside andmise aeg lepitakse kokku tööde käigus taasesitamist võimaldavas vormis (Tellija ja Täitja poolt);
- 5.2.3. Tellija korraldab tulemite üleandmise järel vajalikud lähtekoodi paigaldused keskkondadesse;
- 5.2.4. Täitja viib läbi testkeskkonnas suitsutesti;
- 5.2.5. Tellija viib läbi vastuvõtutestimise (Acceptance Test, User Acceptance Test);
- 5.2.6. 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.

## 5.3. Dokumentatsioon

- 5.3.1. Vastavalt alapunktis 4 kirjeldatud arendustööde skoobile tuleb töö käigus luua ja täiendada STARi dokumentatsiooni, milleks on minimaalselt:
  - 5.3.1.1. 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;
  - 5.3.1.2. Tehniline dokumentatsioon (nt paigaldusjuhendid Confluence keskkonnas);
  - 5.3.1.3. Andmemudel;
  - 5.3.1.4. Kasutusjuhend;
- 5.3.2. Dokumentatsiooni täpsed liigid ja sisu formaat lepitakse kokku tööde tellimise käigus.

## 5.4. Nõuded automaattestimisele

- 5.4.1. Loodav lähtekood peab olema kaetud 75% ulatuses ühiktestidega, 50% ulatuses integratsiooni/liideste/süsteemi/"end-to-end" automatiseeritud testidega;
- 5.4.2. Ühiktestid käivitatakse enne koodi kokku ehitamist;
- 5.4.3. Ühiktestid on üks osa üleantavast koodist;
- 5.4.4. Testidega kaetakse serveripoolne ja kasutajaliidese kood, kui arendaja ei lepi TEHIKu arhitektiga kokku teisiti;
- 5.4.5. Funktsionalsed automaattestide tegemine lepitakse Tellija Täitja poolt kokku tööde käigus;
- 5.4.6. Tellija ja täitja peavad koostöös panema arenduskeskkonnas olevad automaattestid (API testid) tööle ka vähemalt testkeskkonnas või kui on kokkulepitud, siis ka teistes olemasolevates keskkondades;
- 5.4.7. Automatiseeritud testiliikide haldamisega ja ajakohasena hoidmisega tegeleb arendaja;
- 5.4.8. Automaattestide raporteerimiseks seatakse TEHIKu koodihoidla keskkonnas (Gitlab) üles pideva integratsiooni töövoog, mis seadistatakse SonarQube koodi ja testide kaetuse analüsaatoriga koos toimima;



- 5.4.9. SonarQube's kontrollitavate vigade ja haavatavuse taseme kontroll peab olema edukalt läbitud;
- 5.4.10. Kõik testide automatiseerimise tööriistad/raamistikud peavad oskama töötada GitLab PipeLine's ja Docker Container's;
- 5.4.11. Liideste testimisel tuleb lisaks positiivsele töövoole katta ka negatiivseid töövooge;
- 5.4.12. Liideste testide lähtekood peab olema kättesaadav GitLabis;
- 5.4.13. Automaattestid peavad üle antavas lähtekoodis pideva integratsiooni töövoos edukalt läbima;
- 5.4.14. Arendaja peab hoolt kandma, et varasemalt loodud testid jäävad töö lõpetamisel edukalt läbitavaks;
- 5.4.15. Nõuete muutumisel peab arendaja viima testid vastavusse uute nõuetega või neid eemaldama, kui selliseid nõudeid enam ei eksisteeri;
- 5.4.16. Mõistlik ulatus kasutajaliidese testimisest peab olema automatiseeritud ja korduvkasutatav tööde käigus kokkulepitud raamistikul;
- 5.4.17. Automaattestimise (API test) artefaktidena luuakse:
  - 5.4.17.1. Paigaldusjuhend automaattestide integreerimiseks testkeskkonda ja teistesse kokkulepitud keskkondadesse;
  - 5.4.17.2. Kasutusjuhend automaattestide käivitamiseks.

## **5.5. Koormustestimised**

- 5.5.1. Rakendus peab toetama paralleelselt 500 päringut (siinkohal on oodatav arendajapoolne hinnang, kas see on piisav päringute maht, vajadusel tõsta arvu);
- 5.5.2. API teenustele ning keskmine päringu teenindamise aeg ei tohi olla suurem kui 3 sekundit (peab hindama API päringu keerukust). Projekti käigus täpsustatakse vastamiseajad;
- 5.5.3. Nõude täitmise kontrolliks tuleb luua koormustestid. Koormustestid peavad olema juhitud muutujatega ehk seadistatavad. Koormustestide tegemine lepib Tellija ja Taitja poolt kokku tööde käigus.

## **5.6. Mittefunktsionaalsed nõuded**

- 5.6.1. Mittefunktsionaalsete nõuete täitmisel tuleb lähtuda järgnevast:
  - 5.6.1.1. Uue funktsionaalsuse loomisel tuleb lähtuda raamlepingus ja selle lisades kirjeldatud nõuetest;
  - 5.6.1.2. Põhjendatud juhtudel võib seatud nõuetest kõrvale kalduda taasesitamist võimaldavas vormis sõlmitud kokkuleppel tellijaga.